Convolutional Networks

Lecture 03

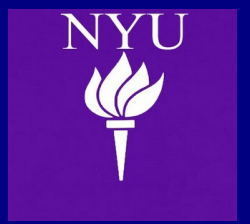
Yann Le Cun

Facebook AI Research,

Center for Data Science, NYU

Courant Institute of Mathematical Sciences, NYU

http://yann.lecun.com

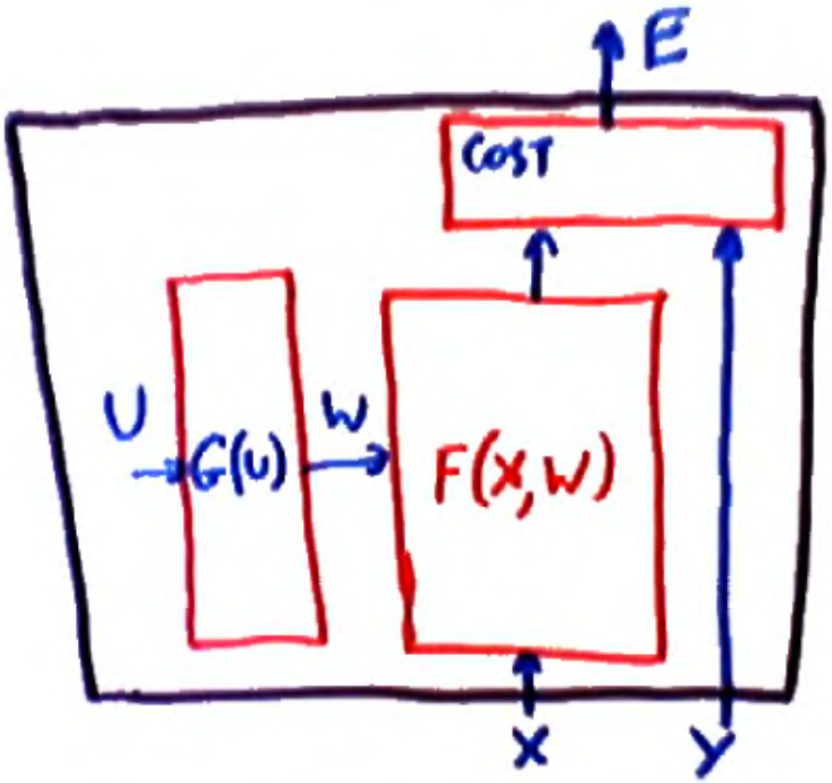



Parameter Transforms
In
Deep Networks

Parameter Space Transform

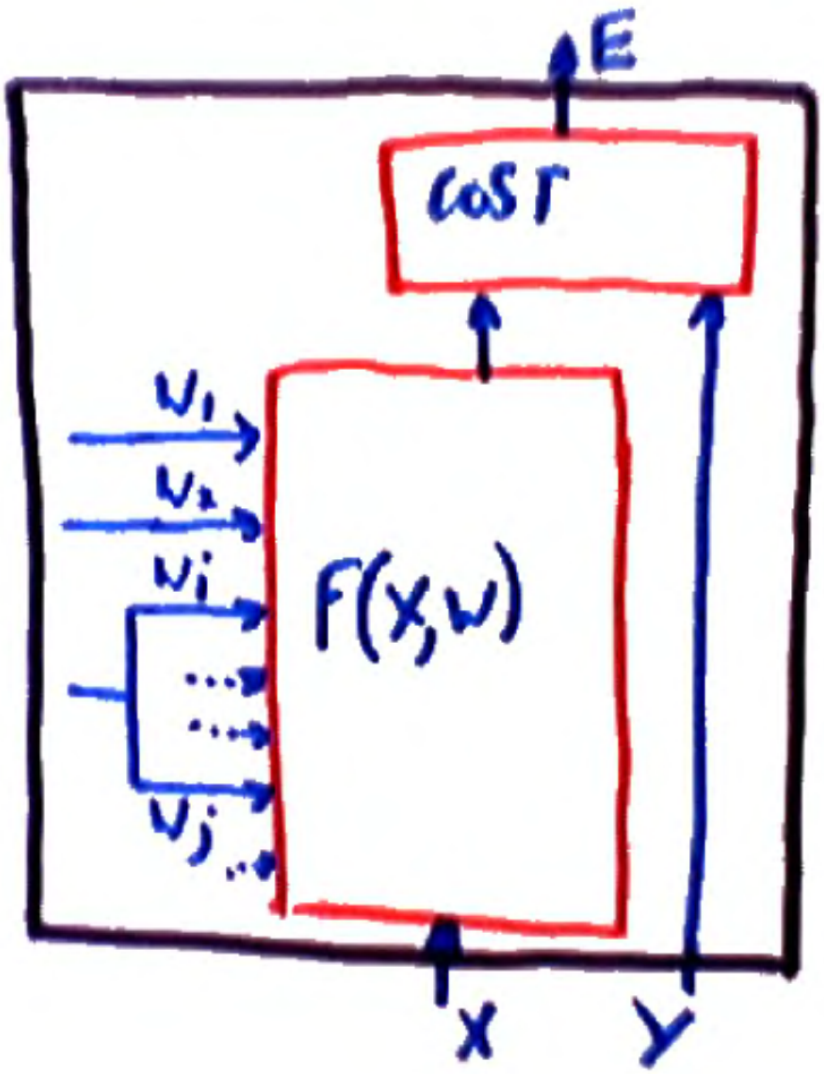
Reparameterizing the function by transforming the space

$$E(Y, X, W) \rightarrow E(Y, X, G(U))$$



- gradient descent in U space:
  $U \leftarrow U - \eta \frac{\partial G}{\partial U}' \frac{\partial E(Y,X,W)}{\partial W}'$
- equivalent to the following algorithm in W space: $W \leftarrow W - \eta \frac{\partial G}{\partial U} \frac{\partial G}{\partial U}' \frac{\partial E(Y,X,W)}{\partial W}'$
- dimensions: $[N_w \times N_u][N_u \times N_w][N_w]$

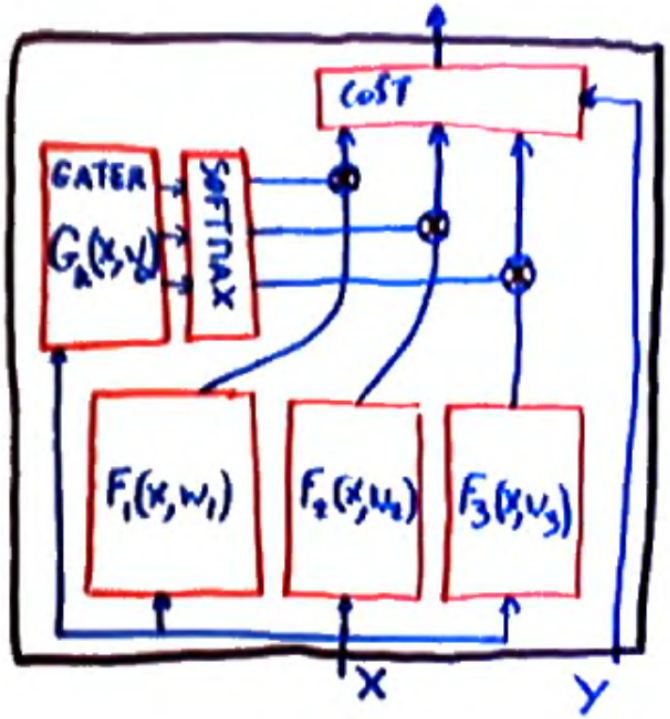
Parameter Space Transform: Weight Sharing



- A single parameter is replicated multiple times in a machine
- $E(Y, X, w_1, \ldots, w_i, \ldots, w_j, \ldots) \rightarrow E(Y, X, w_1, \ldots, u_k, \ldots, u_k, \ldots)$
- gradient: $\frac{\partial E()}{\partial u_k} = \frac{\partial E()}{\partial w_i} + \frac{\partial E()}{\partial w_j}$
- w_i and w_j are tied, or equivalently, u_k is shared between two locations.

Mixture of Experts

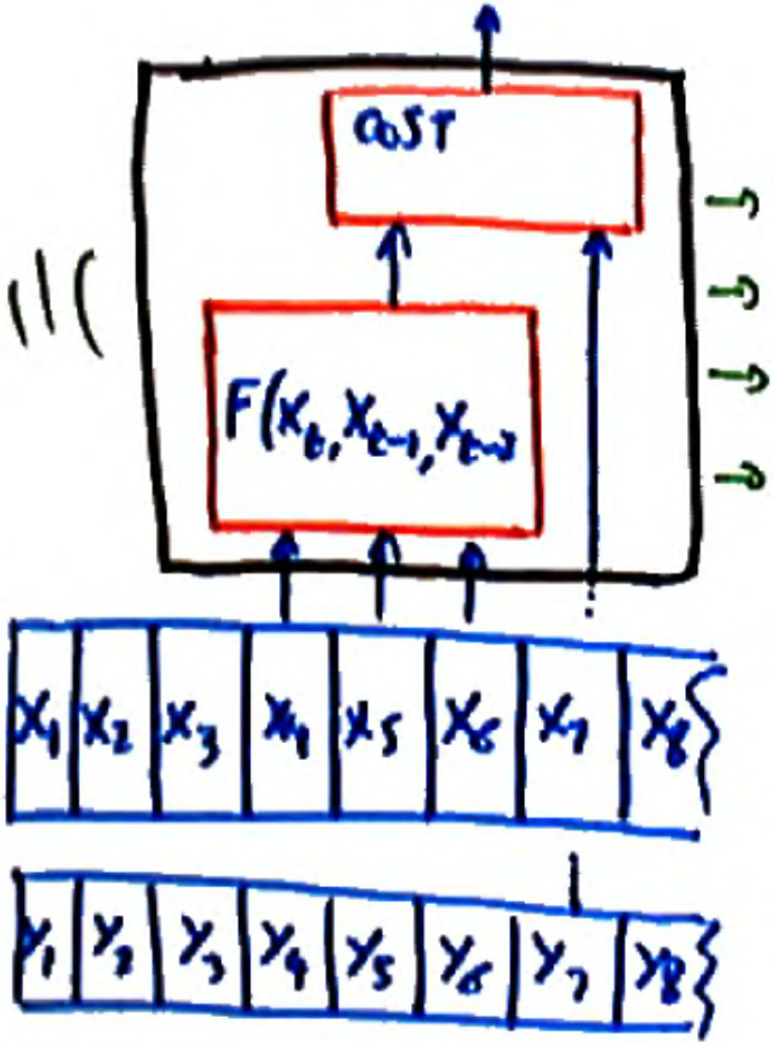
Sometimes, the function to be learned is consistent in restricted domains of the input space, but globally inconsistent. Example: piecewise linearly separable function.



- Solution: a machine composed of several "experts" that are specialized on subdomains of the input space.
- The output is a weighted combination of the outputs of each expert. The weights are produced by a "gater" network that identifies which subdomain the input vector is in.
- $F(X,W) = \sum_k u_k F^k(X,W^k)$ with
  $u_k = \frac{\exp(-\beta G_k(X,W^0))}{\sum_k \exp(-\beta G_k(X,W^0))}$
- the expert weights u_k are obtained by softmax-ing the outputs of the gater.
- example: the two experts are linear regressors, the gater is a logistic regressor.

Time-Delayed Inputs

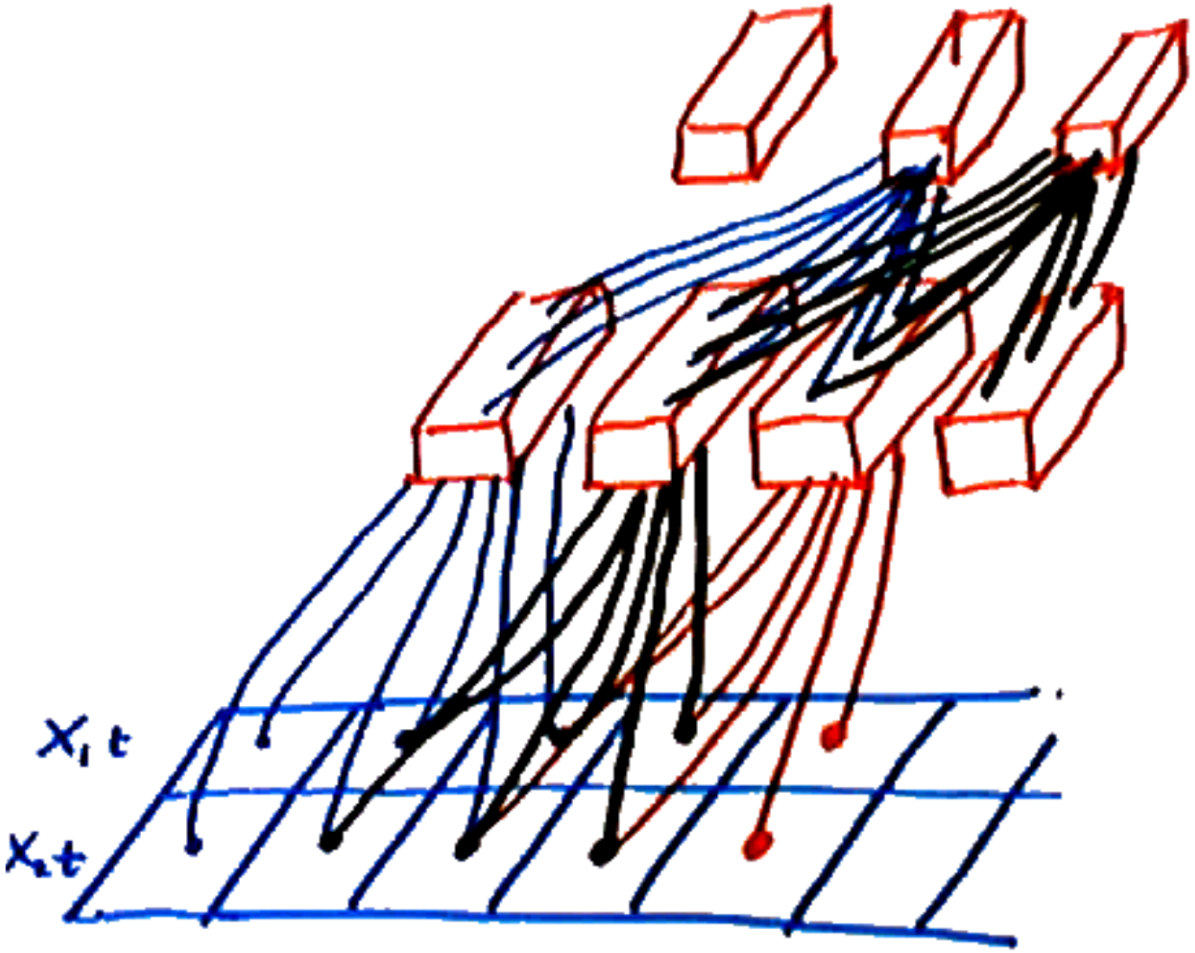
The input is a sequence of vectors X_t.



- simple idea: the machine takes a time window as input
- $R = F(X_t, X_{t-1}, X_{t-2}, W)$
- Examples of use:
  - predict the next sample in a time series (e.g. stock market, water consumption)
  - predict the next character or word in a text
  - classify an intron/exon transition in a DNA sequence

Example: 1D (Temporal) convolutional net

- 1D (Temporal) ConvNet, aka Timed-Delay Neural Nets
- Groups of units are replicated at each time step.
- Replicas have identical (shared) weights.



Convolutional Networks

(ConvNet or CNN)

Deep Learning = The Entire Machine is Trainable

Traditional Pattern Recognition: Fixed/Handcrafted Feature Extractor

Feature Extractor → Trainable Classifier

Mainstream Modern Pattern Recognition: Unsupervised mid-level features

Feature Extractor → Mid-Level Features → Trainable Classifier

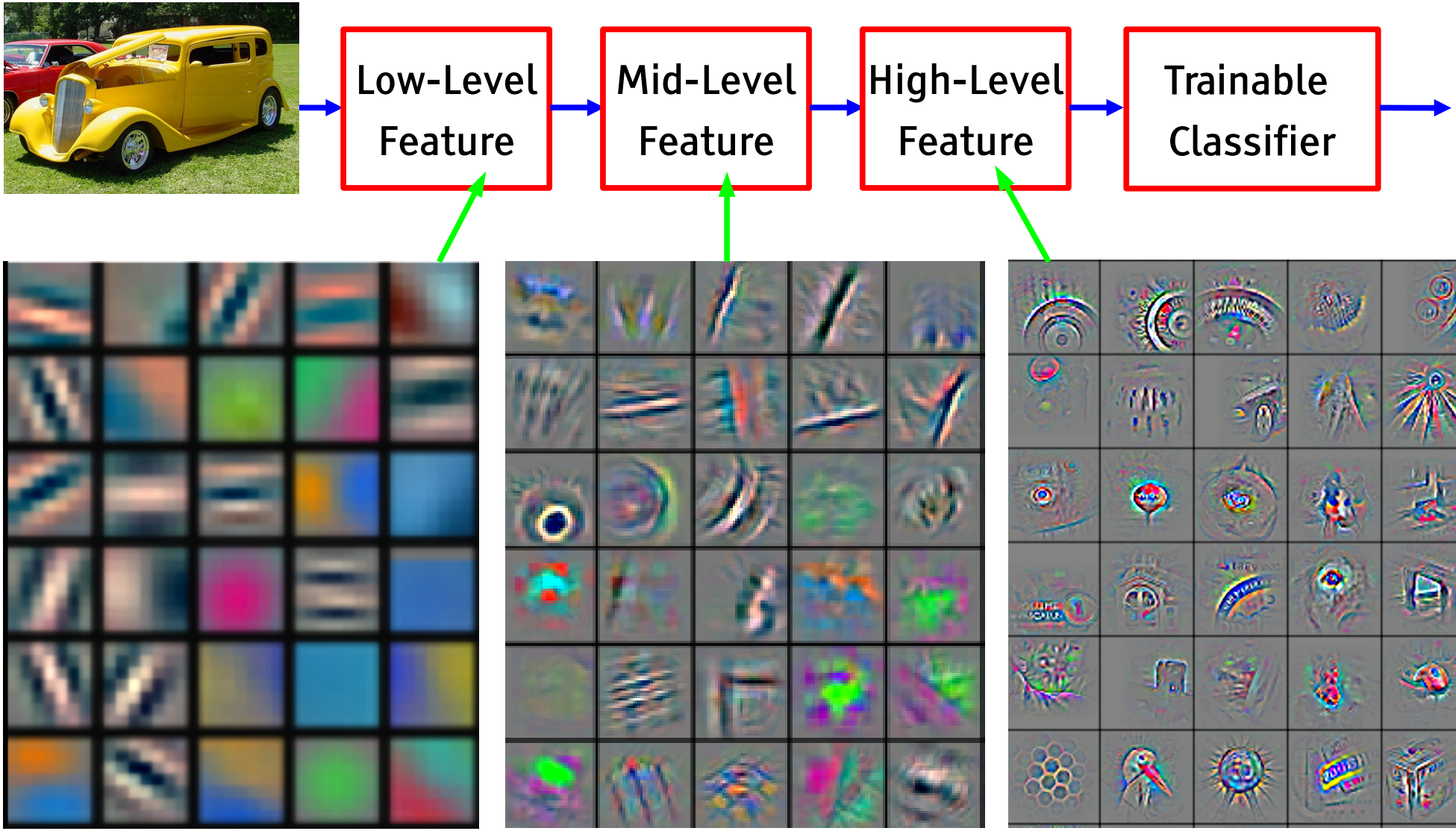
Deep Learning: Representations are hierarchical and trained

Low-Level Features → Mid-Level Features → High-Level Features → Trainable Classifier

Deep Learning = Learning Hierarchical Representations

Y LeCun

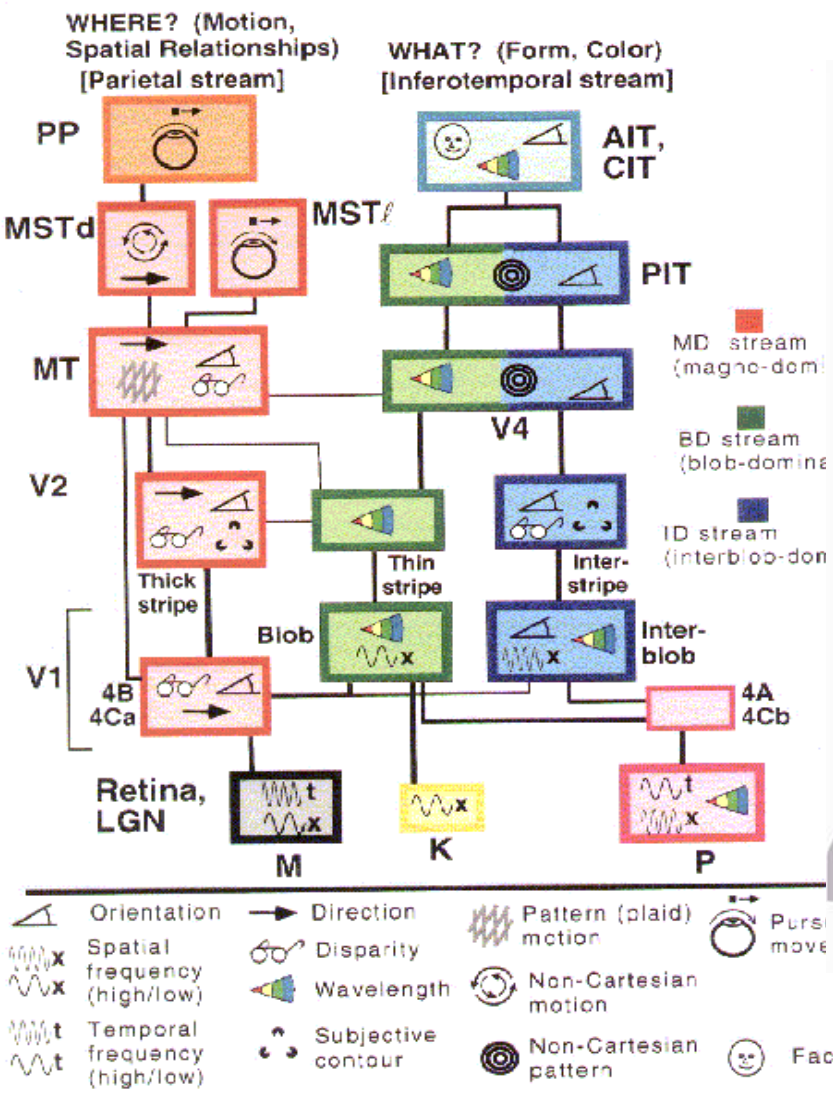
- It's deep if it has more than one stage of non-linear feature transformation

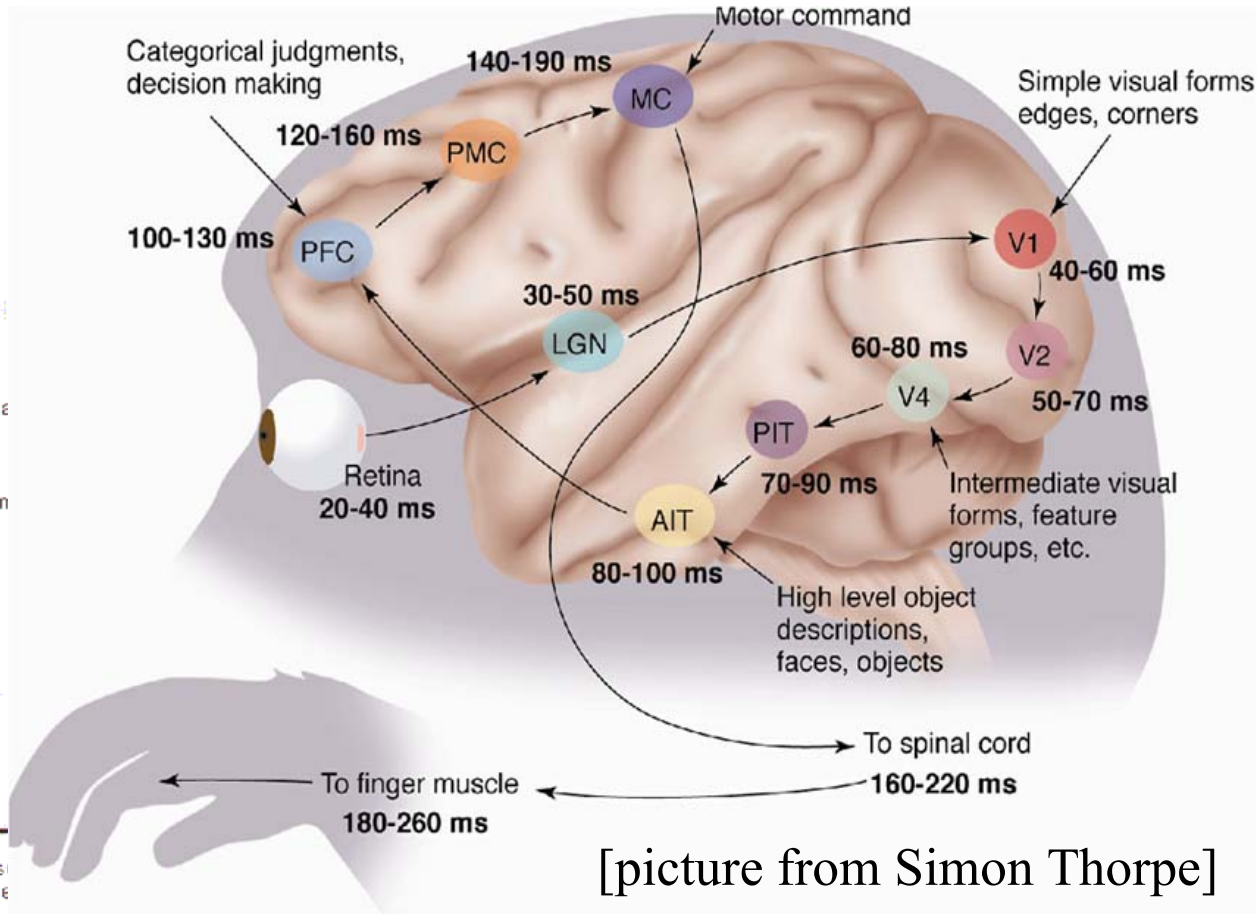


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

How does the brain interprets images?

- The ventral (recognition) pathway in the visual cortex has multiple stages
- Retina - LGN - V1 - V2 - V4 - PIT - AIT ....





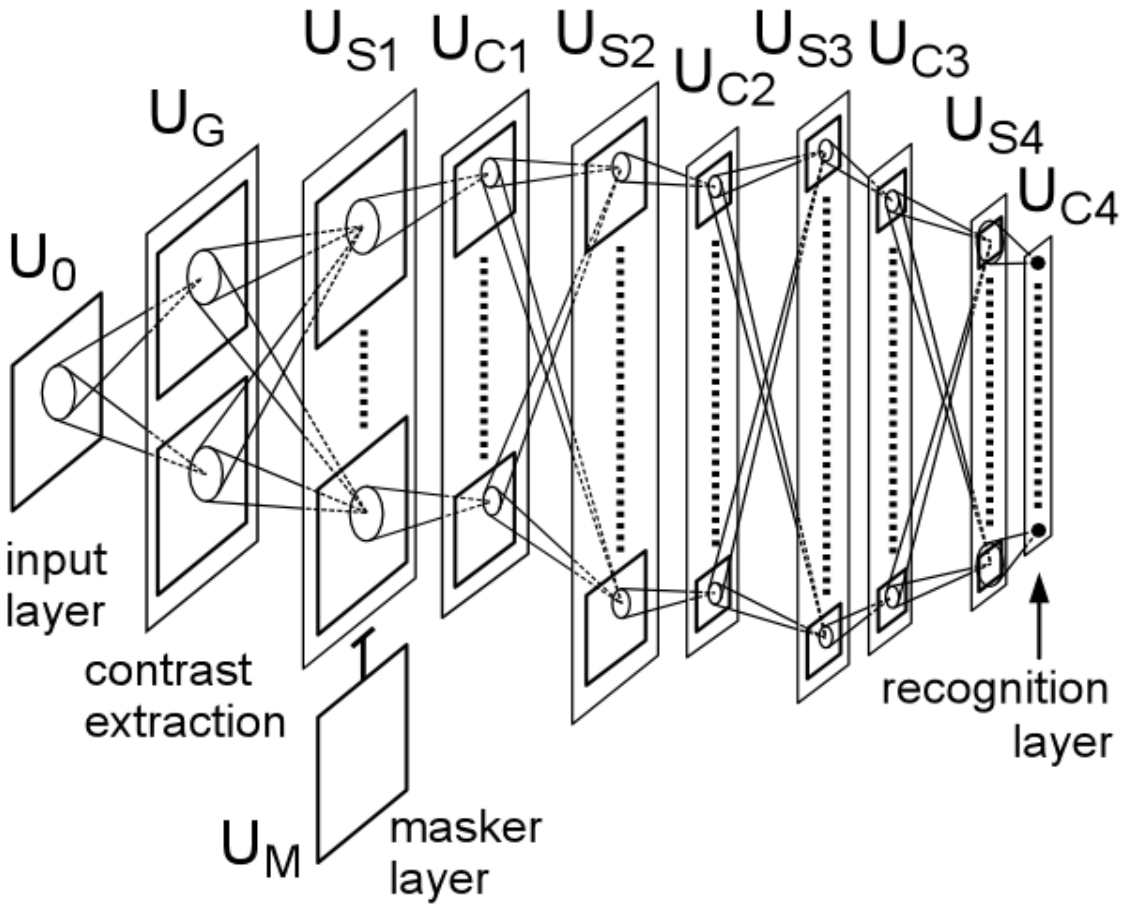
[picture from Simon Thorpe]

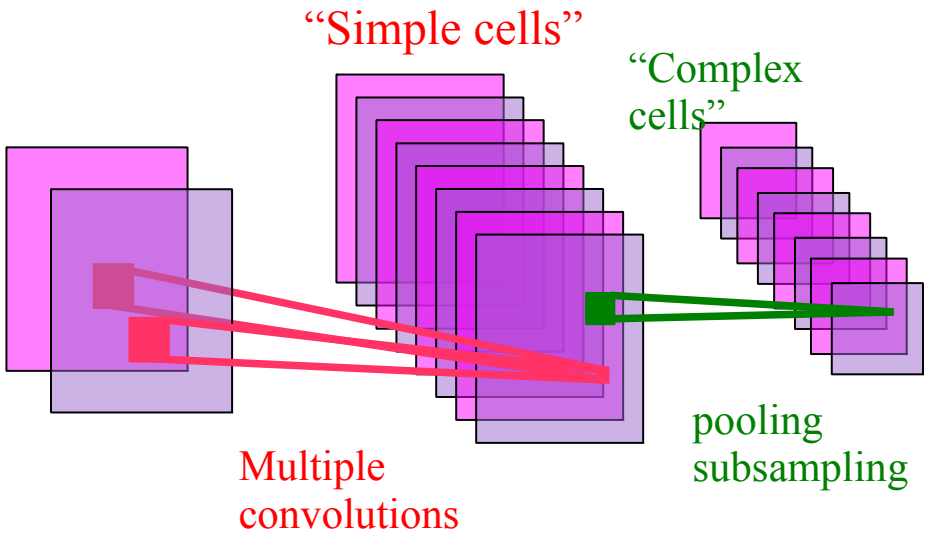
[Gallant & Van Essen]

Hubel & Wiesel's Model of the Architecture of the Visual Cortex

Y LeCun

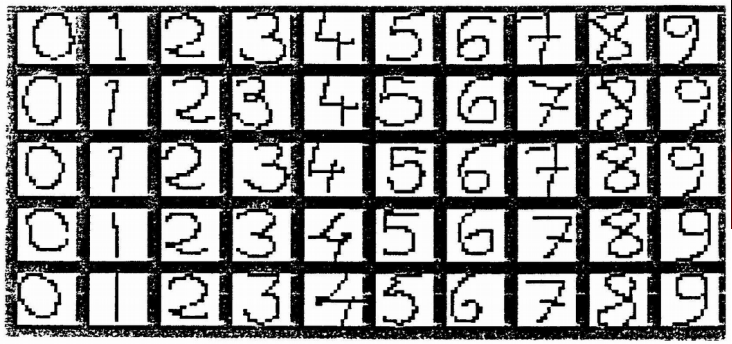
- [Hubel & Wiesel 1962]:
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.



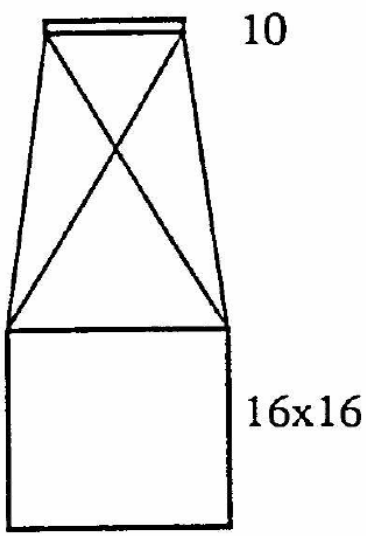


[Fukushima 1982] [LeCun 1989, 1998], [Riesenhuber 1999].....

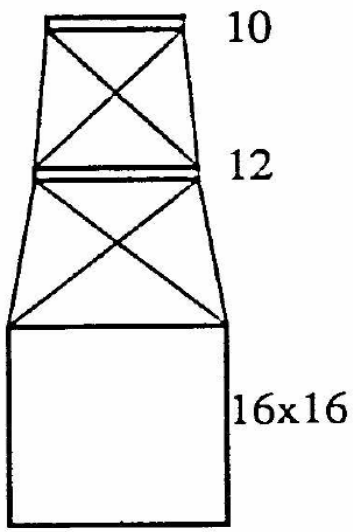
First ConvNets (U Toronto) [LeCun 88, 89]



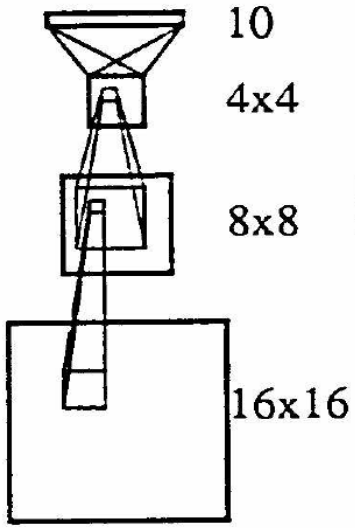
Trained with Backprop. 320 examples.



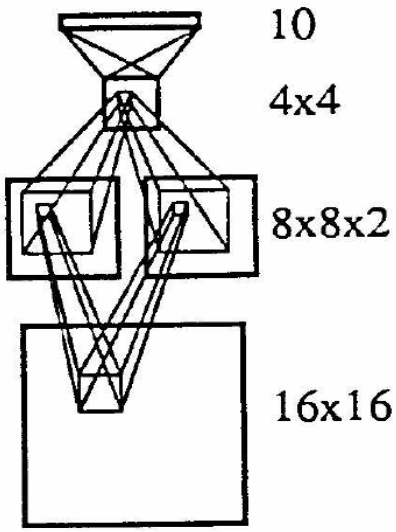
Single layer



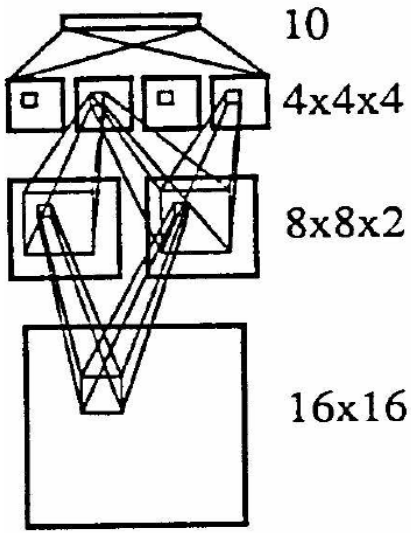
Two layers FC



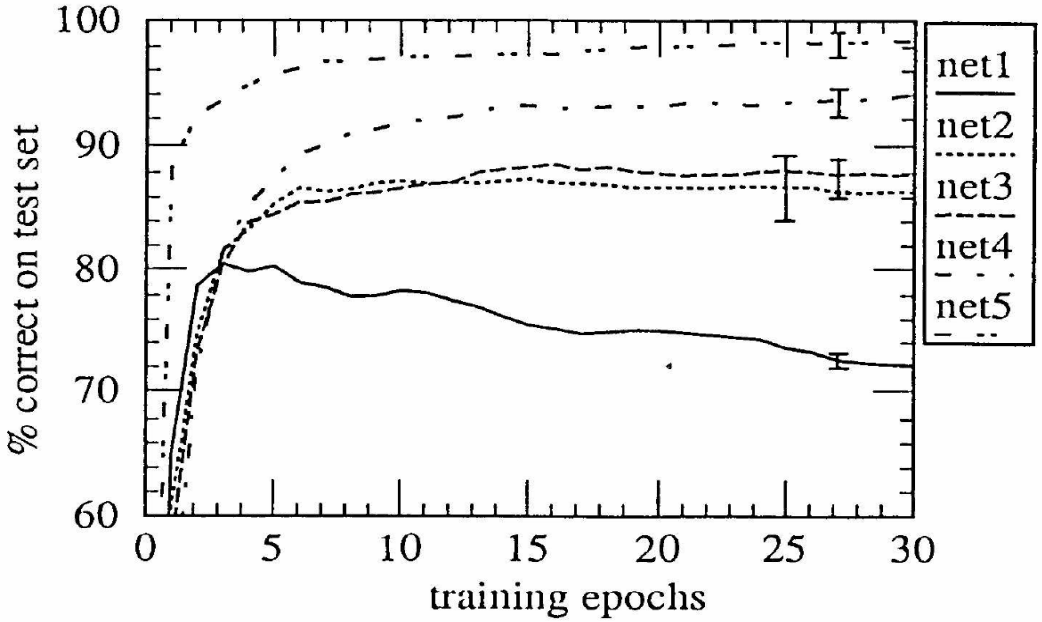
locally connected



Shared weights



Shared weights



- Convolutions with stride (subsampling)
- No separate pooling layers

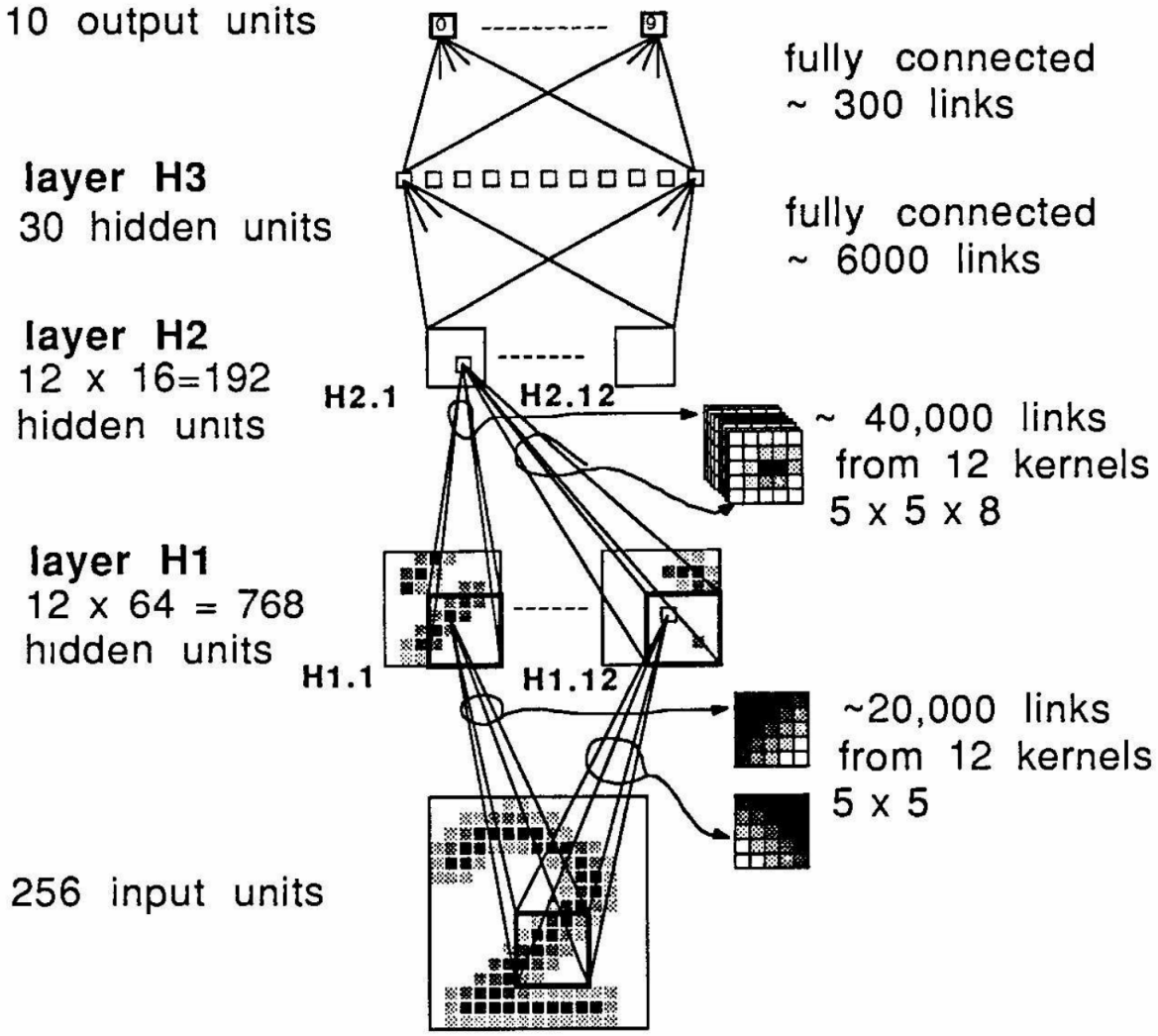
network architecture	links	weights	performance
single layer network	2570	2570	80 %
two layer network	3240	3240	87 %
locally connected	1226	1226	88.5 %
constrained network	2266	1132	94 %
constrained network 2	5194	1060	98.4 %

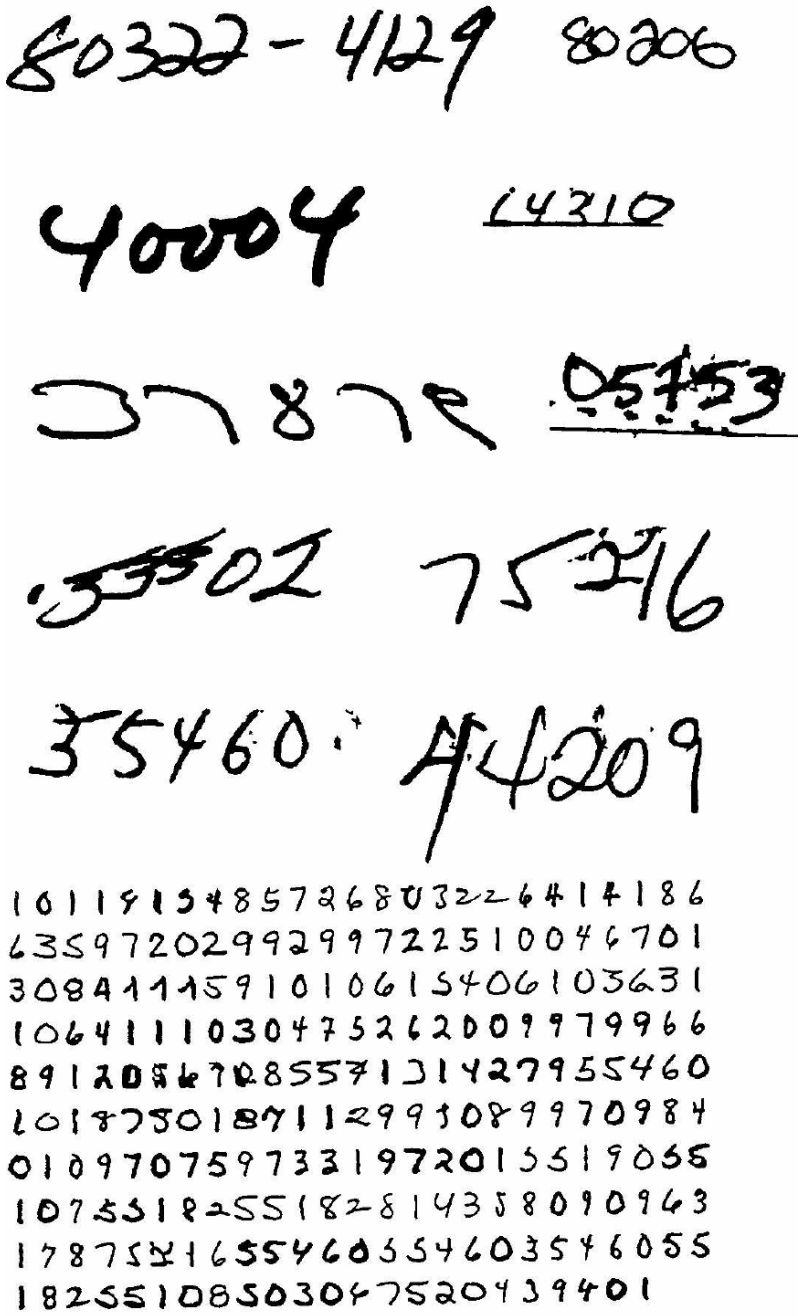
First "Real" ConvNets at Bell Labs [LeCun et al 89]

Trained with Backprop.

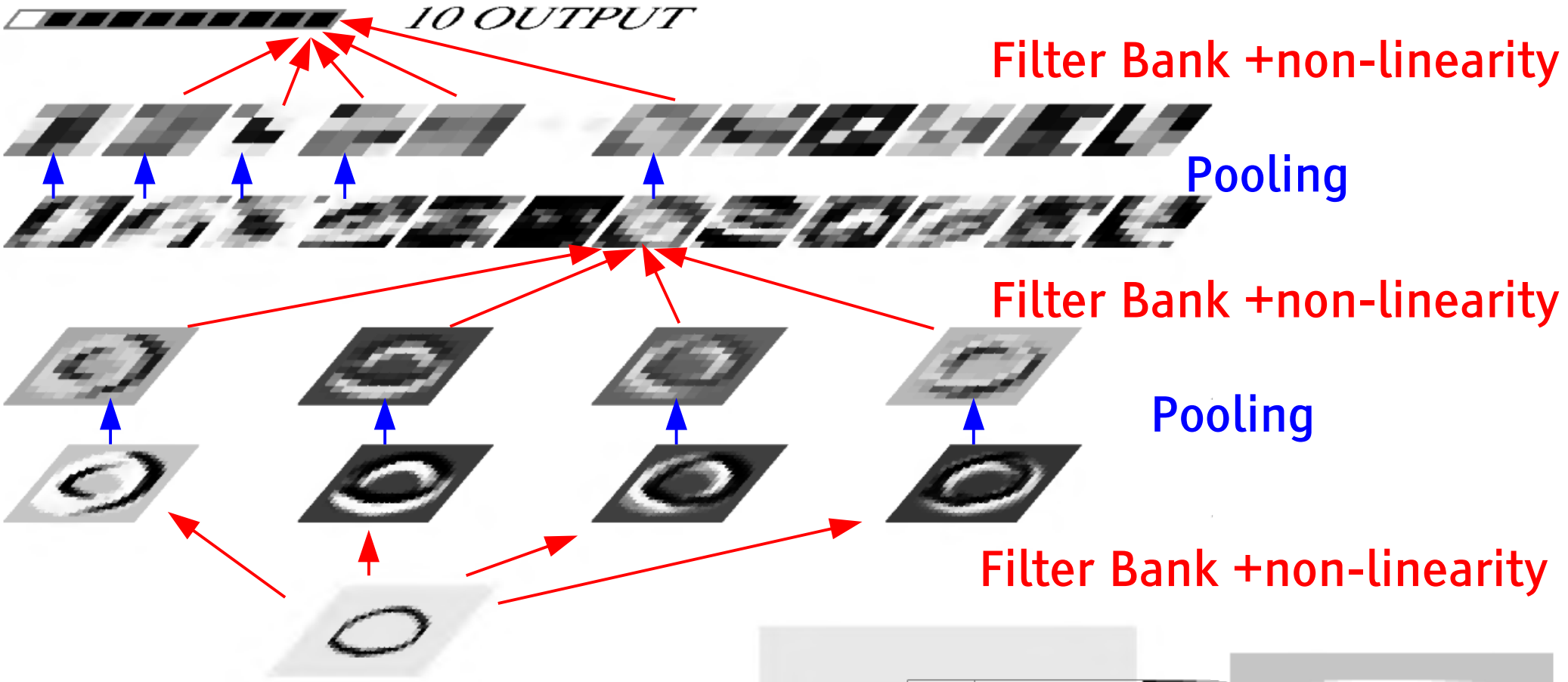
USPS Zipcode digits: 7300 training, 2000 test.

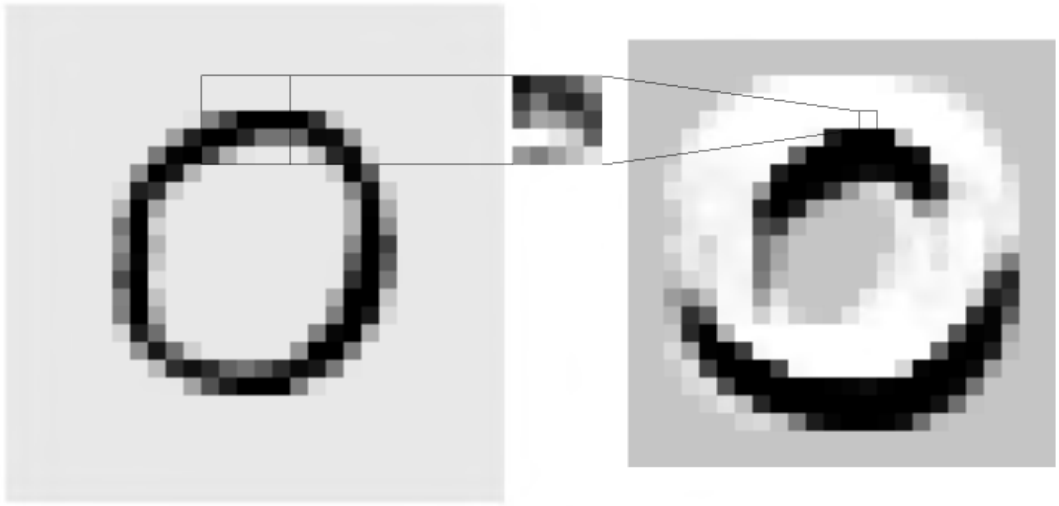
Convolution with stride. No separate pooling.





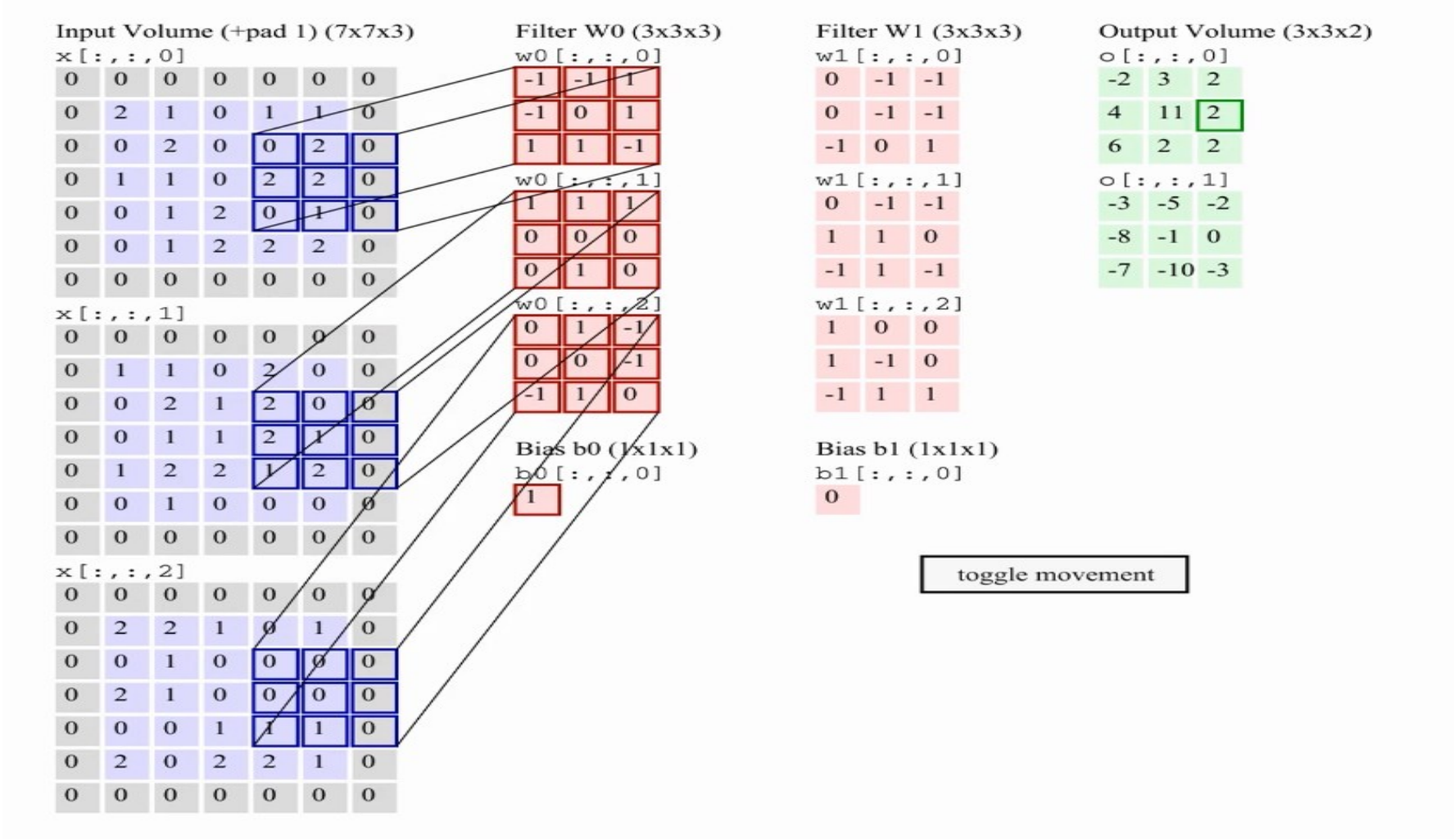
Convolutional Network Architecture





[LeCun et al. NIPS 1989]

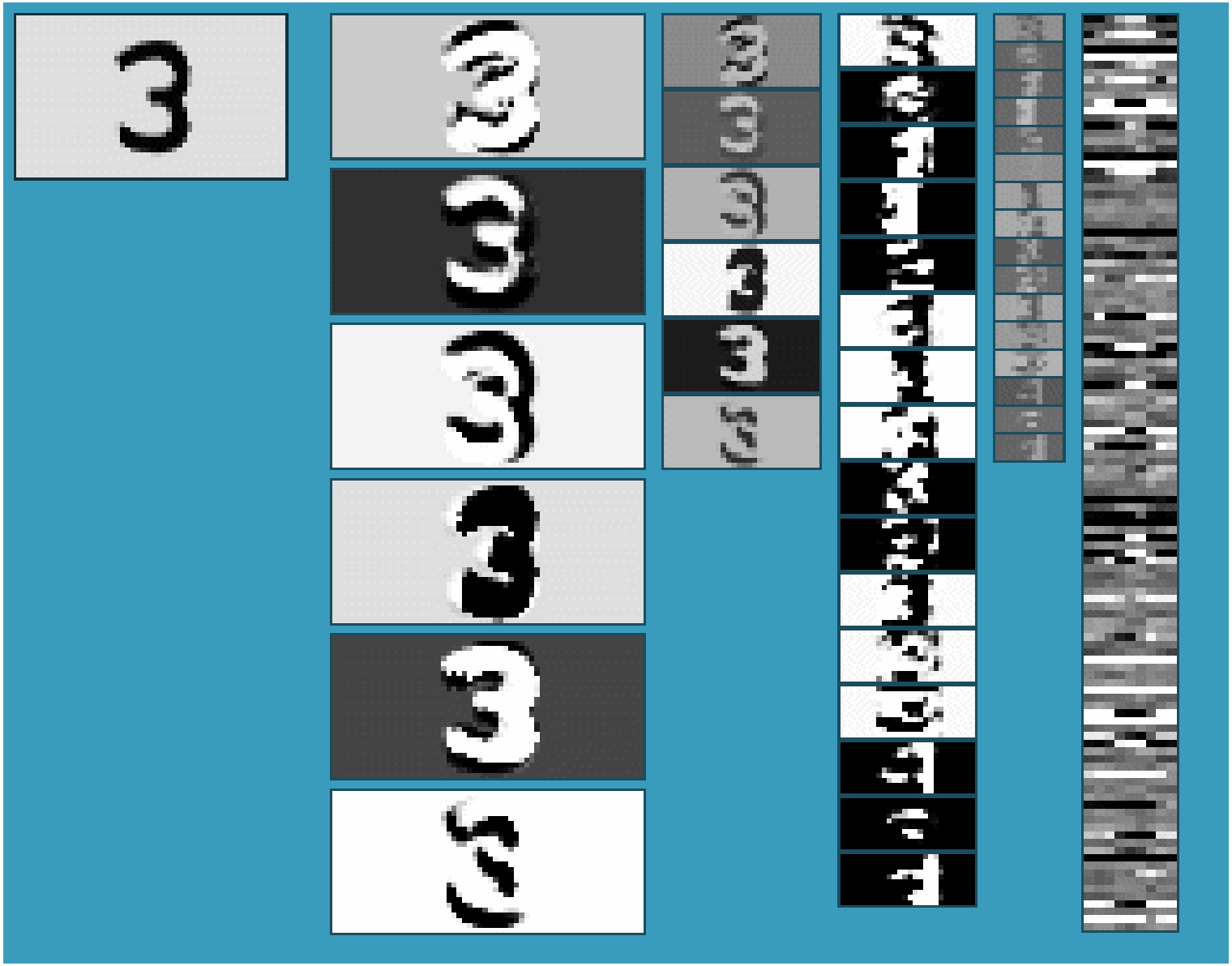
Multiple Convolutions



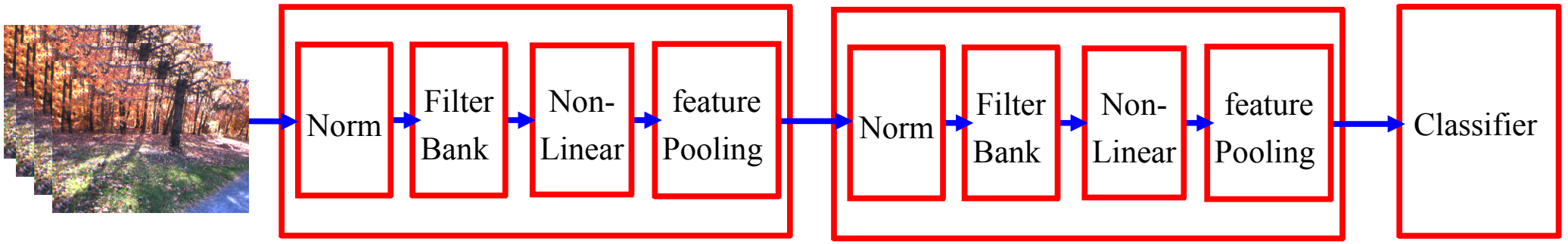
Animation: Andrej Karpathy http://cs231n.github.io/convolutional-networks/

Convolutional Network (vintage 1990)

- Filters-tanh → pooling → filters-tanh → pooling → filters-tanh

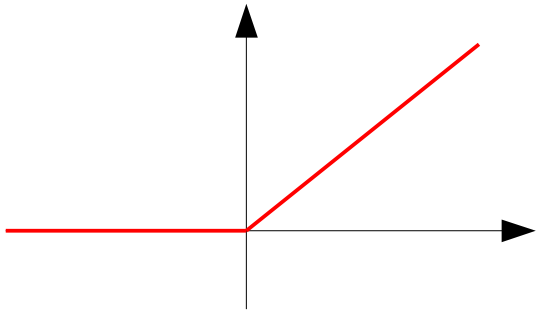


Overall Architecture: multiple stages of Normalization → Filter Bank → Non-Linearity → Pooling



- **Normalization:** variation on whitening (optional)
  - Subtractive: average removal, high pass filtering
  - Divisive: local contrast normalization, variance normalization
- **Filter Bank:** dimension expansion, projection on overcomplete basis
- **Non-Linearity:** sparsification, saturation, lateral inhibition....
  - Rectification (ReLU), Component-wise shrinkage, tanh,..

$$ReLU(x) = max(x, 0)$$



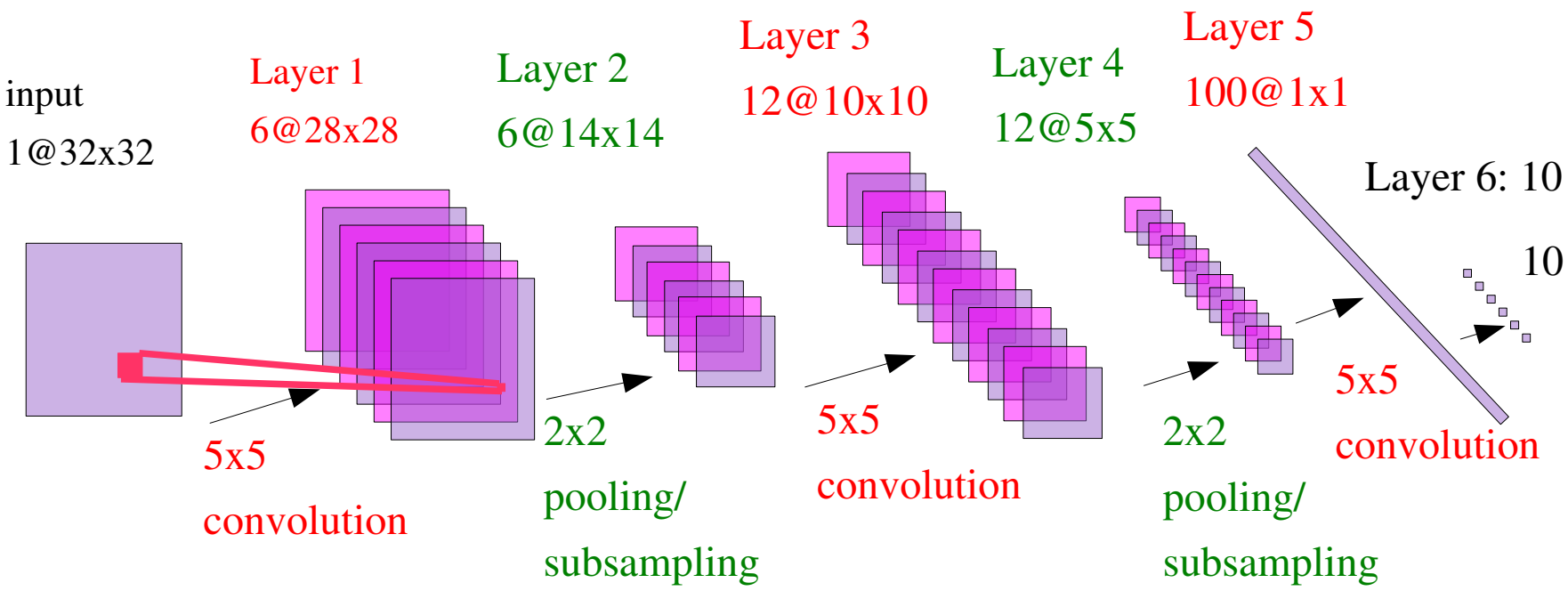
- **Pooling:** aggregation over space or feature type
  - Max, Lp norm, log prob.

$$MAX: Max_i(X_i); \quad L_p: \sqrt[p]{X_i^p}; \quad PROB: \frac{1}{b}\log\left(\sum_i e^{bX_i}\right)$$

LeNet5

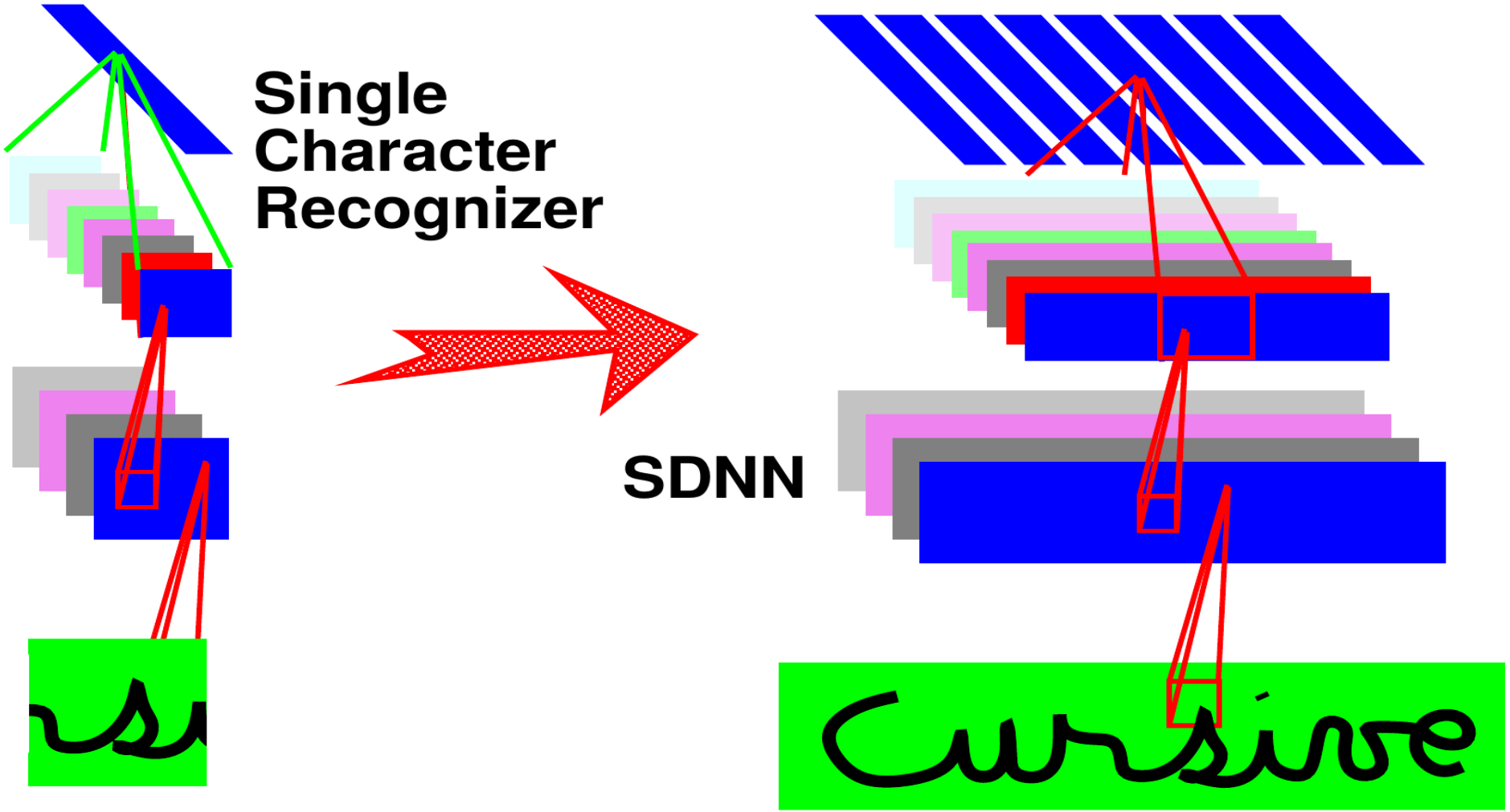
- Simple ConvNet
- for MNIST
- [LeCun 1998]

```
-- LeNet5
-- stage 1 : filter bank -> squashing -> max pooling
model:add(nn.SpatialConvolutionMM(1, 6, 5, 5))
model:add(nn.Tanh())
model:add(nn.SpatialMaxPooling(2, 2, 2, 2))
-- stage 2 : filter bank -> squashing -> max pooling
model:add(nn.SpatialConvolutionMM(6, 12, 5, 5))
model:add(nn.Tanh())
model:add(nn.SpatialMaxPooling(2, 2, 2, 2))
-- stage 3 : standard 2-layer MLP:
model:add(nn.Reshape(12*5*5))
model:add(nn.Linear(12*5*5, 100))
model:add(nn.Tanh())
model:add(nn.Linear(100, nclasses))
```

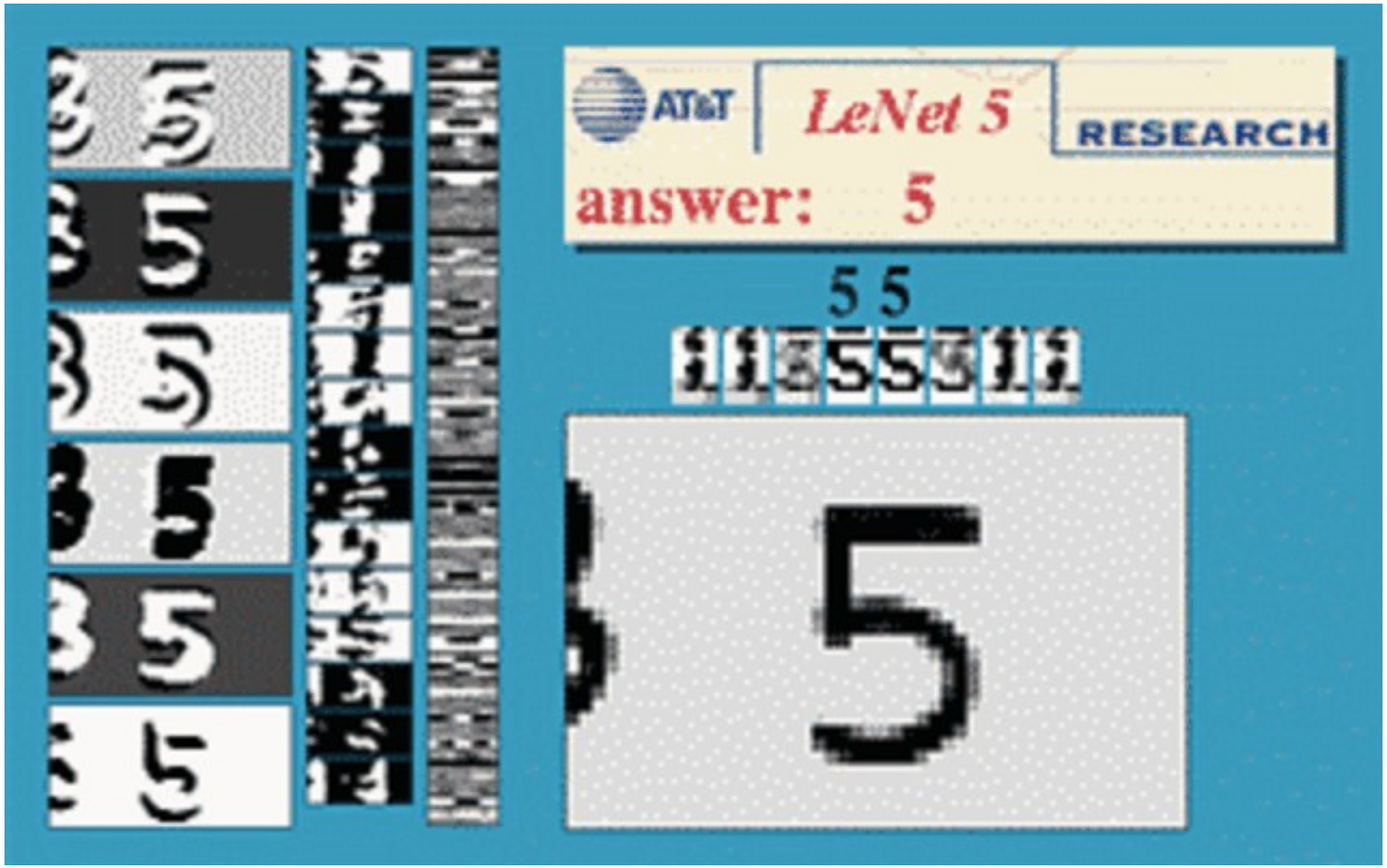


Multiple Character Recognition [Matan et al 1992]

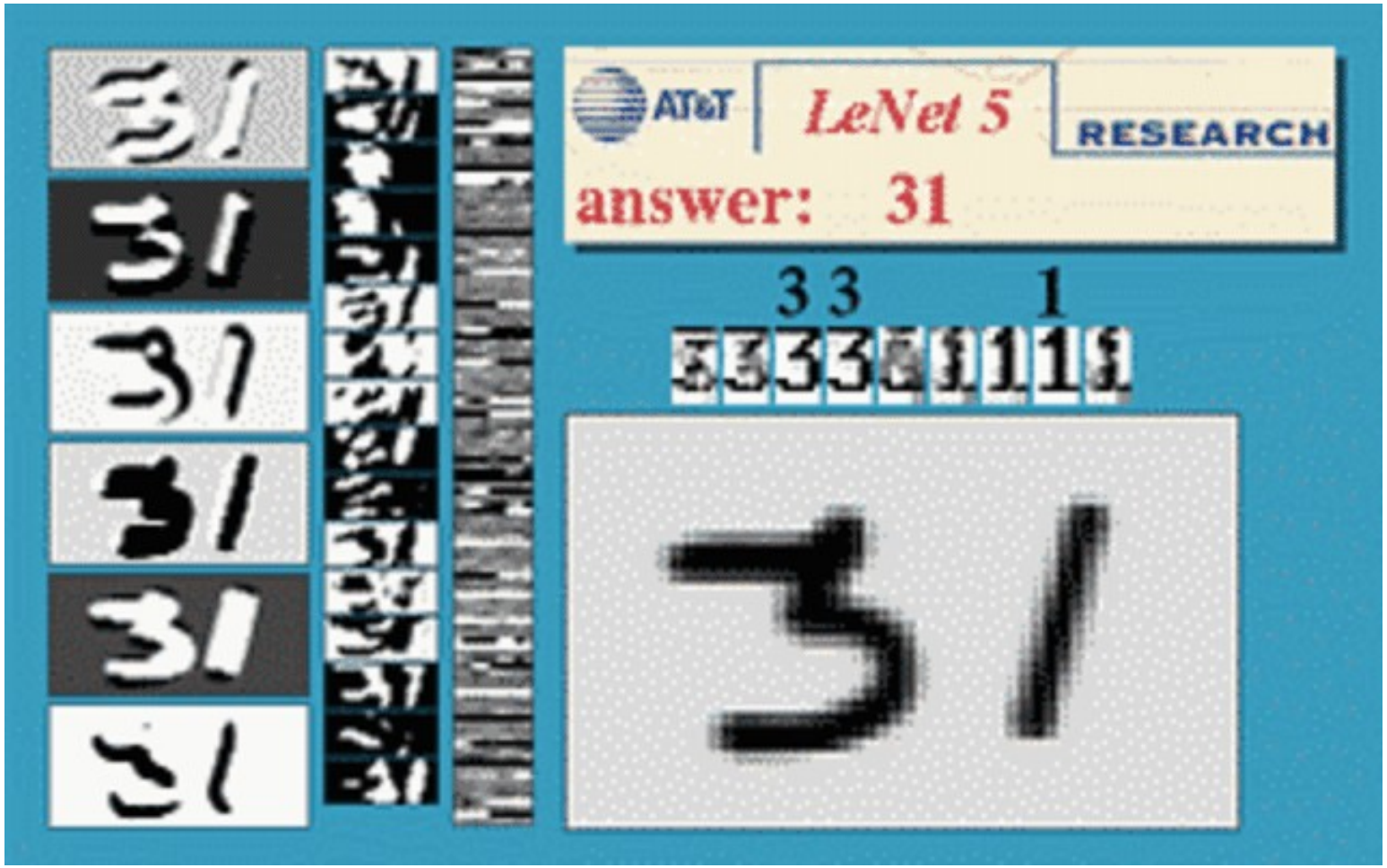
- Every layer is a convolution



Sliding Window ConvNet + Weighted Finite-State Machine



Sliding Window ConvNet + Weighted FSM



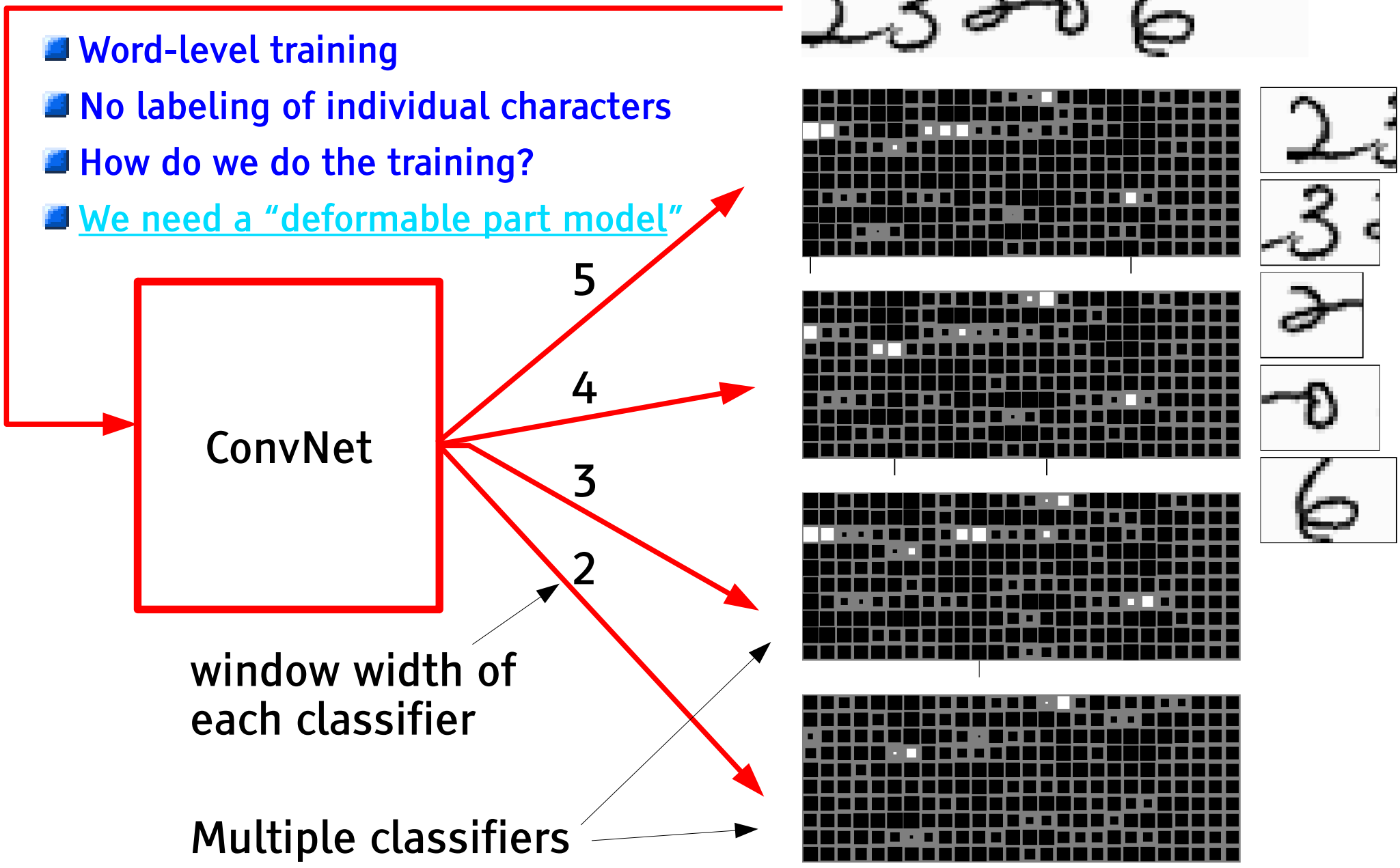
What are ConvNets Good For

- Signals that comes to you in the form of (multidimensional) arrays.
- Signals that have strong local correlations
- Signals where features can appear anywhere
- Signals in which objects are invariant to translations and distortions.

- 1D ConvNets: sequential signals, text
  - Text Classification
  - Musical Genre Recognition
  - Acoustic Modeling for Speech Recognition
  - Time-Series Prediction
- 2D ConvNets: images, time-frequency representations (speech and audio)
  - Object detection, localization, recognition
- 3D ConvNets: video, volumetric images, tomography images
  - Video recognition / understanding
  - Biomedical image analysis
  - Hyperspectral image analysis

Word-level training with weak supervision [Matan et al 1992]

- Word-level training
- No labeling of individual characters
- How do we do the training?
- We need a "deformable part model"



Graph Transformer Networks

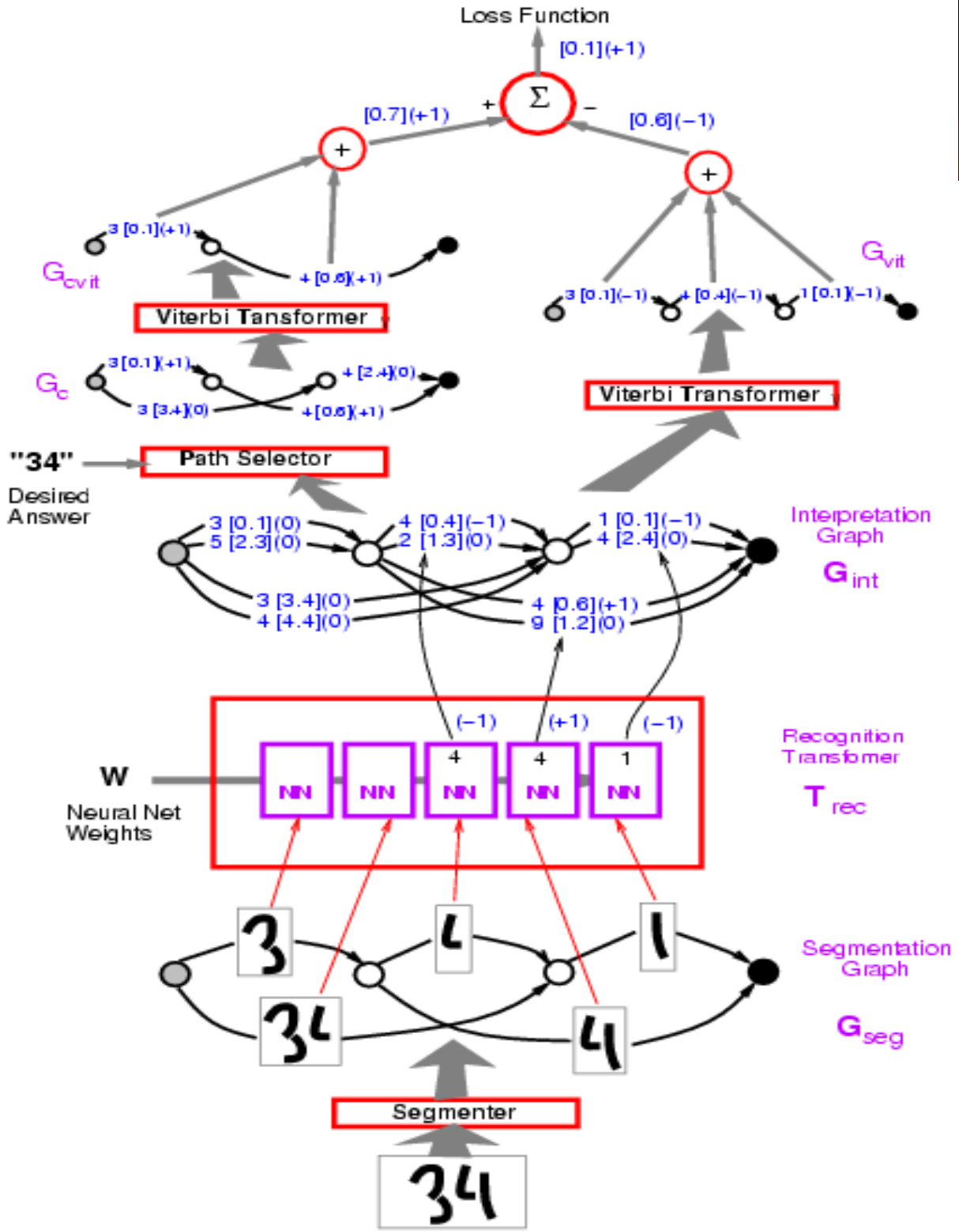
Structured Prediction

on top of Deep Learning

This example shows the structured perceptron loss.

In practice, we used negative log-likelihood loss.

Deployed in 1996 in check reading machines.



Check Reader (Bell Labs, 1995)

Graph transformer network trained to read check amounts.

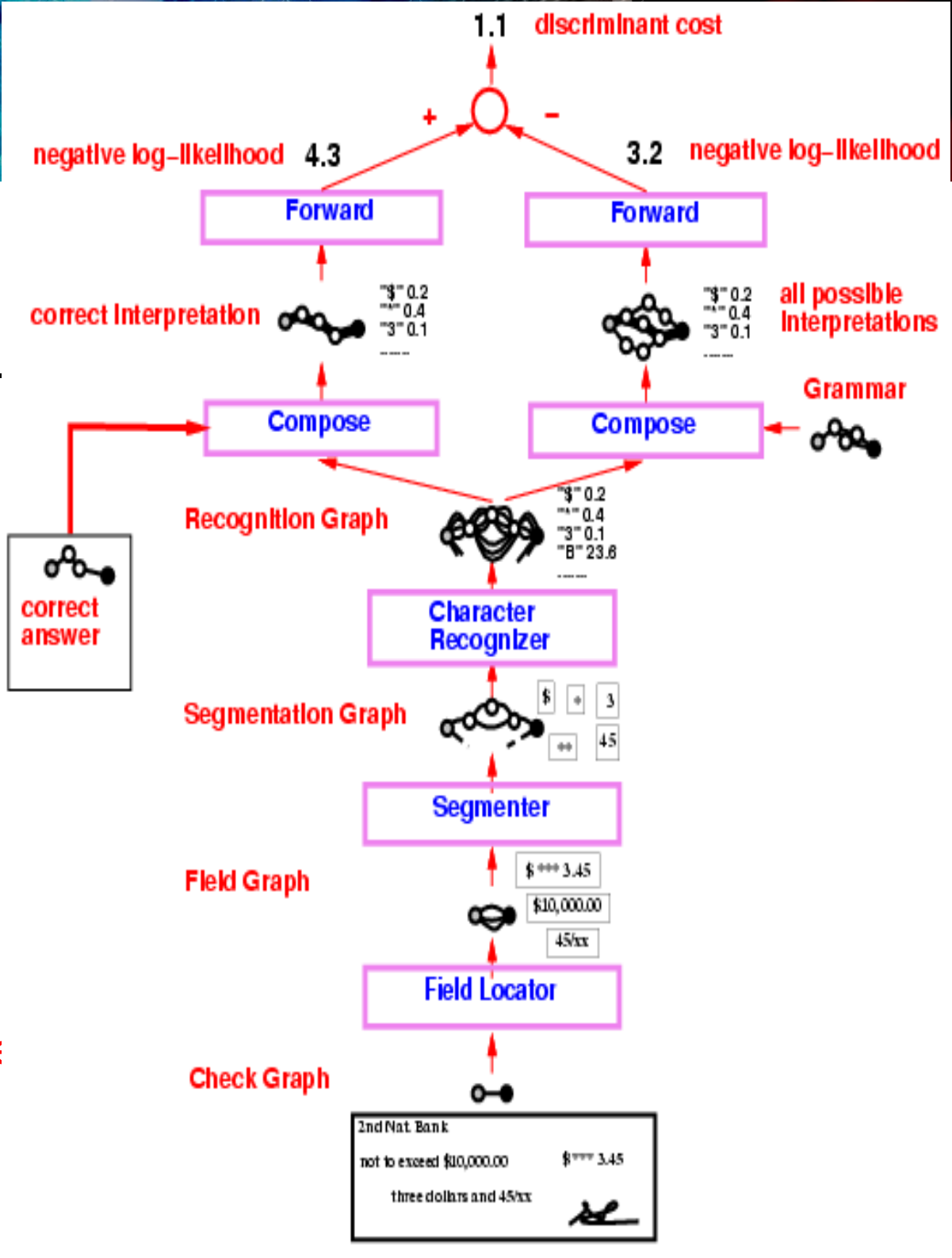
Trained globally with Negative-Log-Likelihood loss.

50% percent correct, 49% reject, 1% error (detectable later in the process).

Fielded in 1996, used in many banks in the US and Europe.

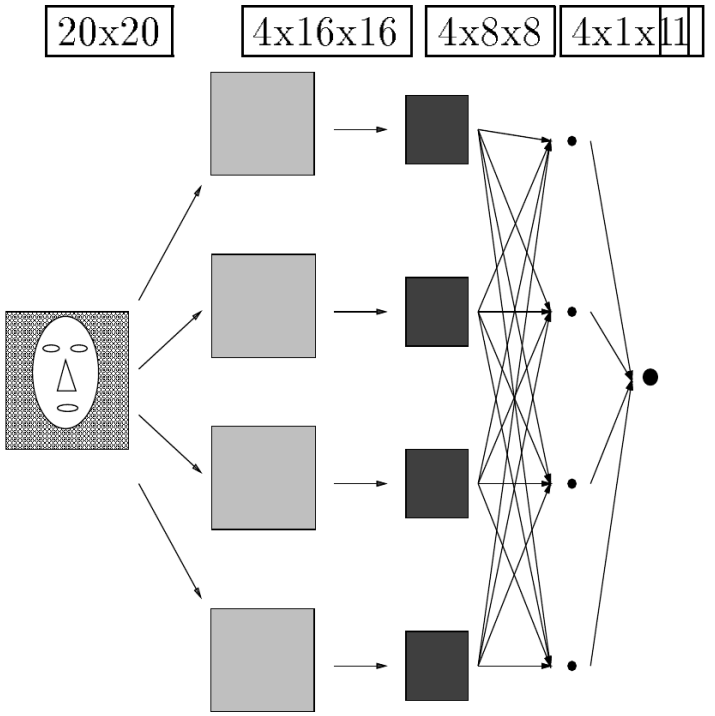
Processed an estimated 10% to 20% of all the checks written in the US in the early 2000s.

[LeCun, Bottou, Bengio, Haffner 1998]



Face Detection [Vaillant et al. 93, 94]

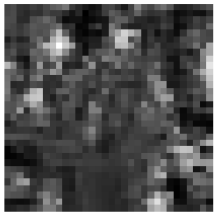
- ConvNet applied to large images
- Heatmaps at multiple scales
- Non-maximum suppression for candidates
- 6 second on a Sparcstation for 256x256 image



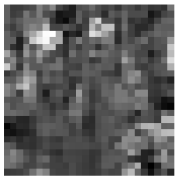
Scale 3

Scale 4

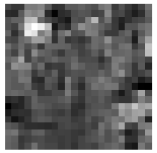
Scale 5



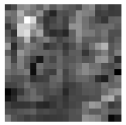
Scale 6



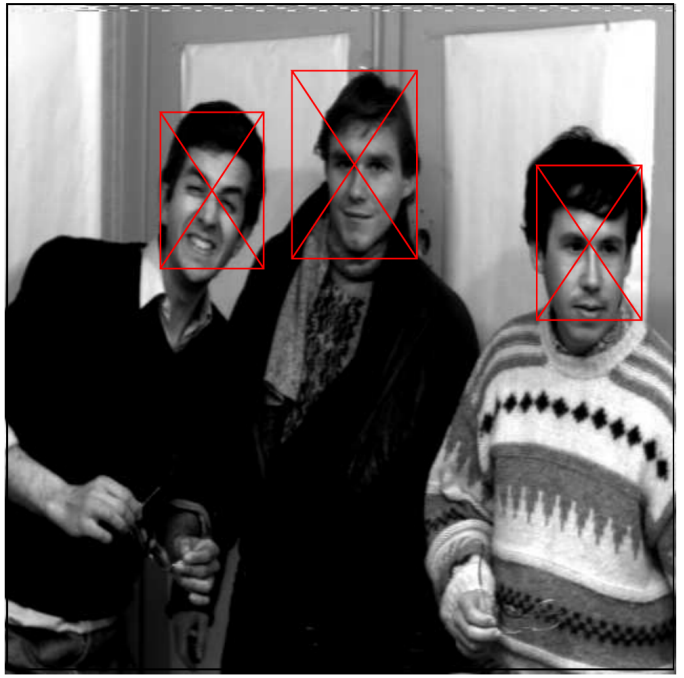
Scale 7



Scale 8

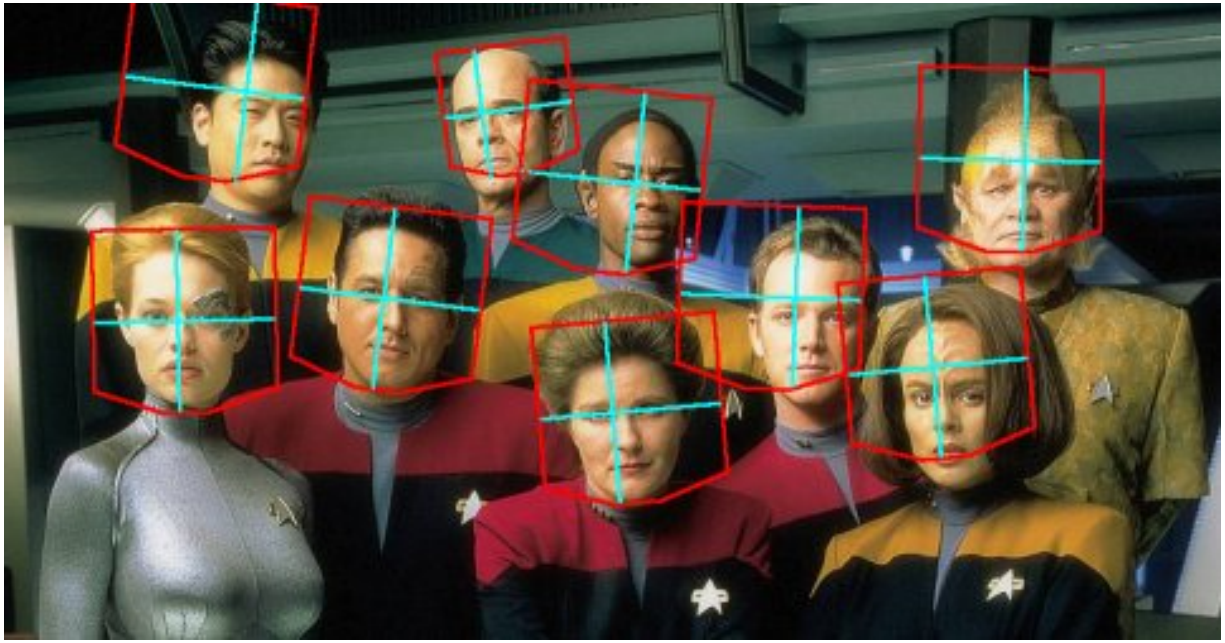


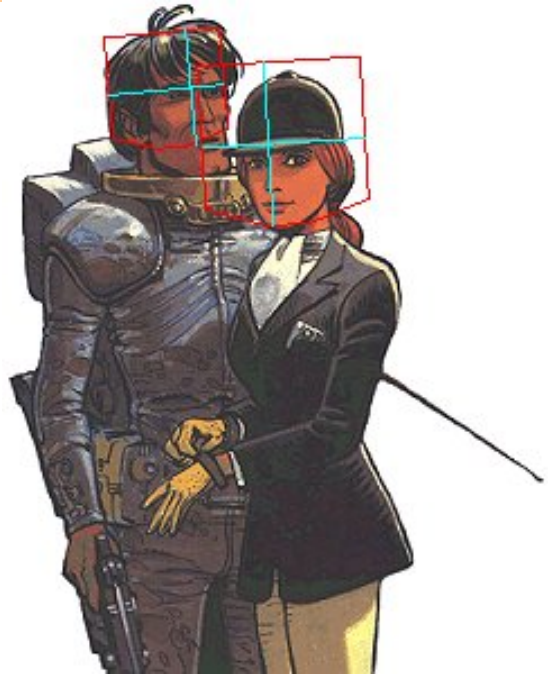
Scale 9



mid 2000s: state of the art results on face detection

Data Set->	TILTED		PROFILE		MIT+CMU	
False positives per image->	4.42	26.9	0.47	3.36	0.5	1.28
Our Detector	90%	97%	67%	83%	83%	88%
Jones & Viola (tilted)	90%	95%	x		x	
Jones & Viola (profile)	x		70%	83%	x	





[Garcia & Delakis 2003] [Osadchy et al. 2004] [Osadchy et al, JMLR 2007]

Simultaneous face detection and pose estimation

Y LeCun

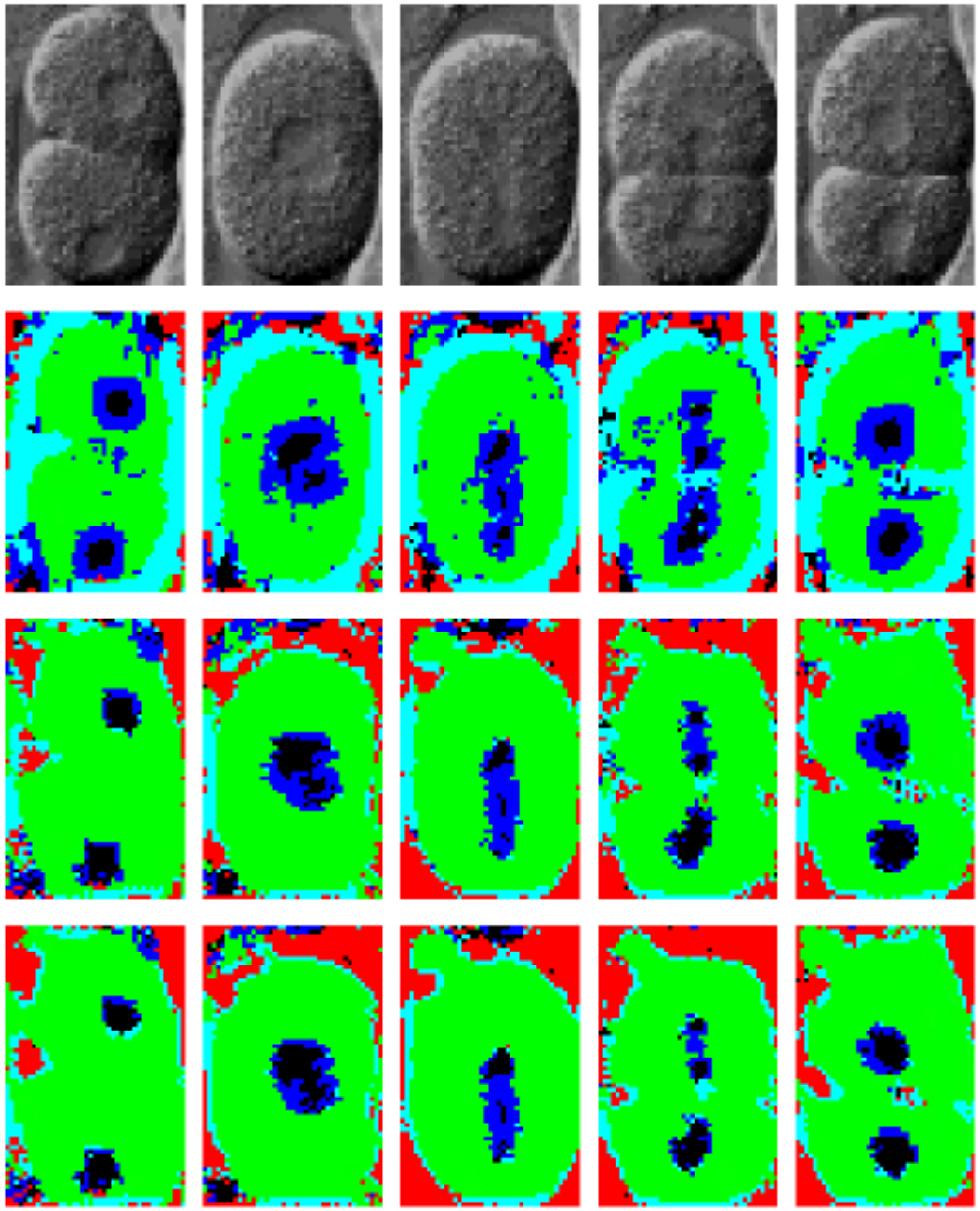
VIDEO: PEDESTRIAN DETECTION

Semantic Segmentation

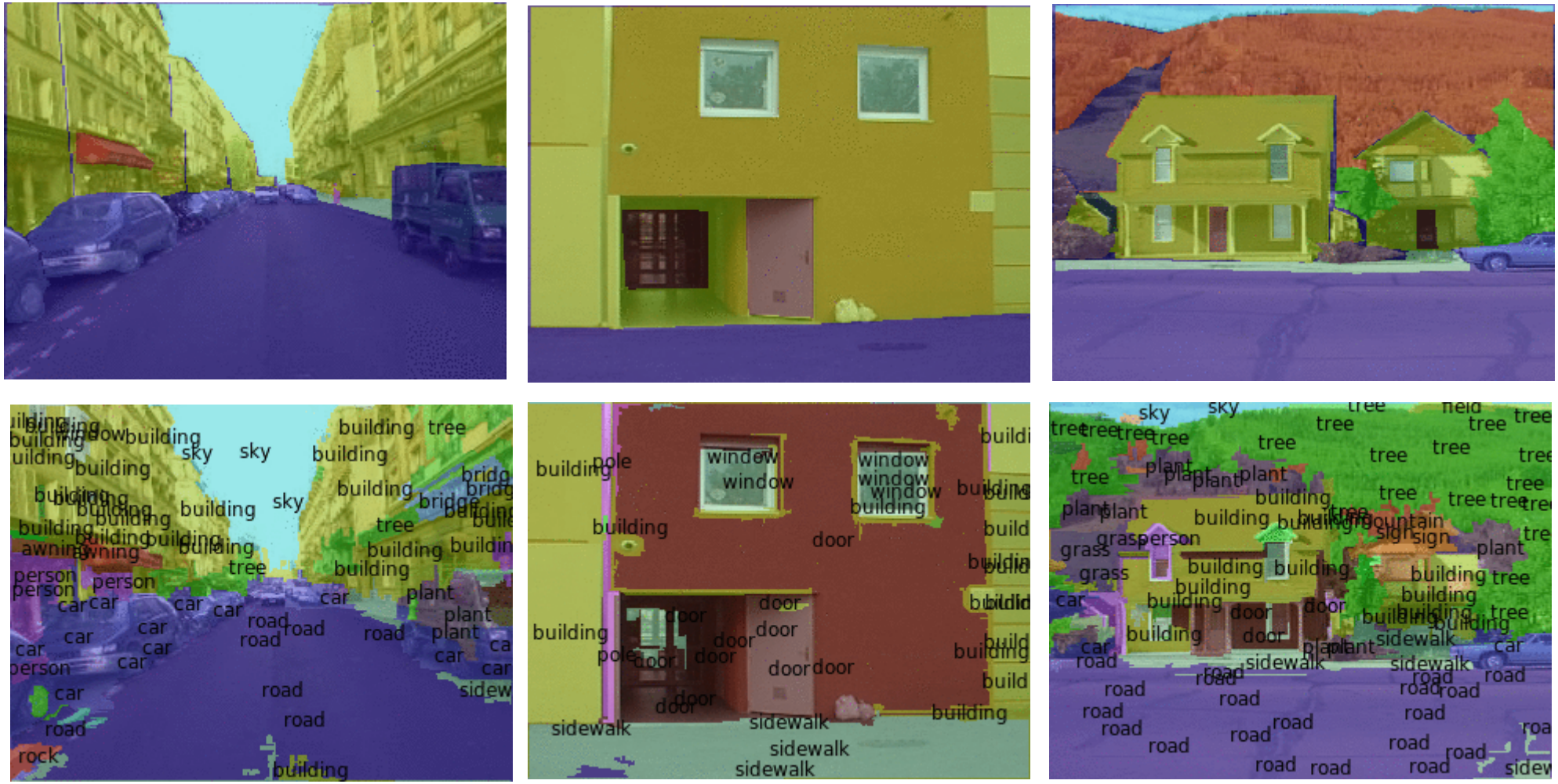
ConvNets for Biological Image Segmentation

Y LeCun

- Biological Image Segmentation
  - [Ning et al. IEEE-TIP 2005]
- Pixel labeling with large context using a convnet
- ConvNet takes a window of pixels and produces a label for the central pixel
- Cleanup using a kind of conditional random field (CRF)
  - Similar to a field of expert, but conditional.
- 3D version for connectomics
  - [Jain et al. 2007]



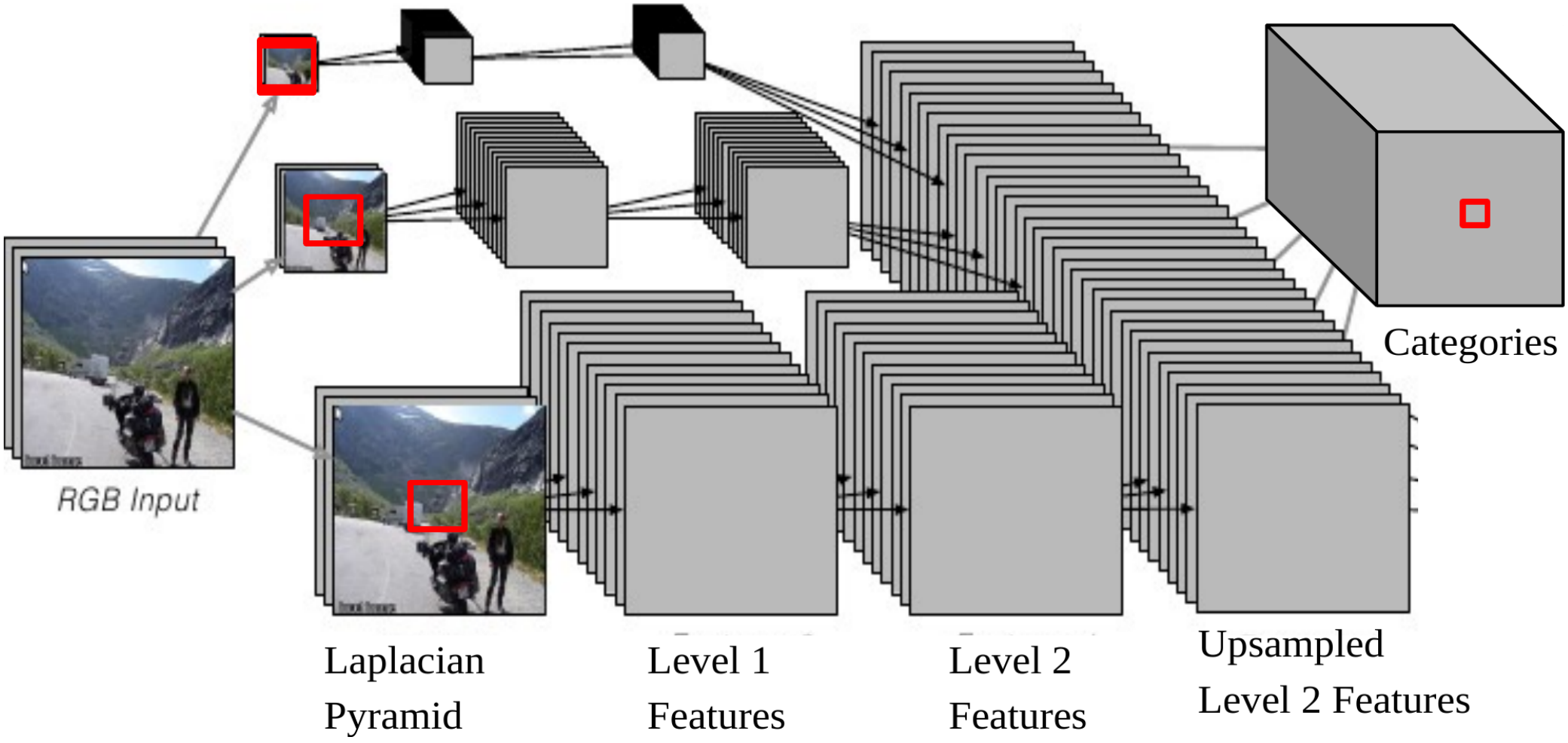
Scene Parsing/Labeling



[Farabet et al. ICML 2012, PAMI 2013]

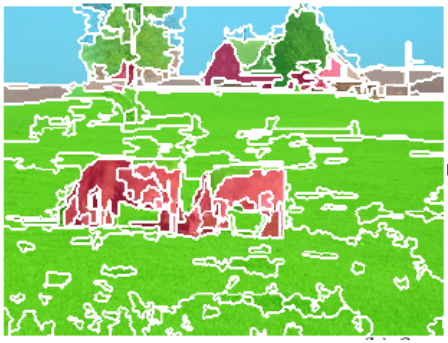
Scene Parsing/Labeling: Multiscale ConvNet Architecture

- Each output sees a large input context:
  - **46x46** window at full rez; **92x92** at ½ rez; **184x184** at ¼ rez
  - [7x7conv]->[2x2pool]->[7x7conv]->[2x2pool]->[7x7conv]->
  - Trained supervised on fully-labeled images



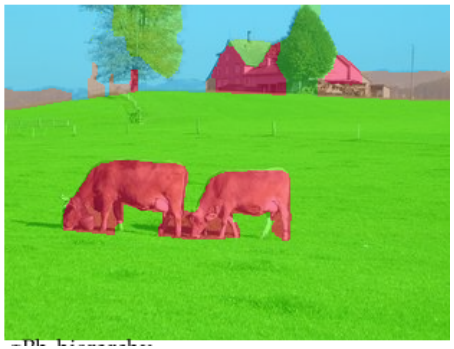
Method 1: majority over super-pixel regions

Super-pixel boundary hypetheses



Superpixel boundaries

Majority Vote Over Superpixels



Categories aligned With region boundaries

Input image

Multi-scale ConvNet

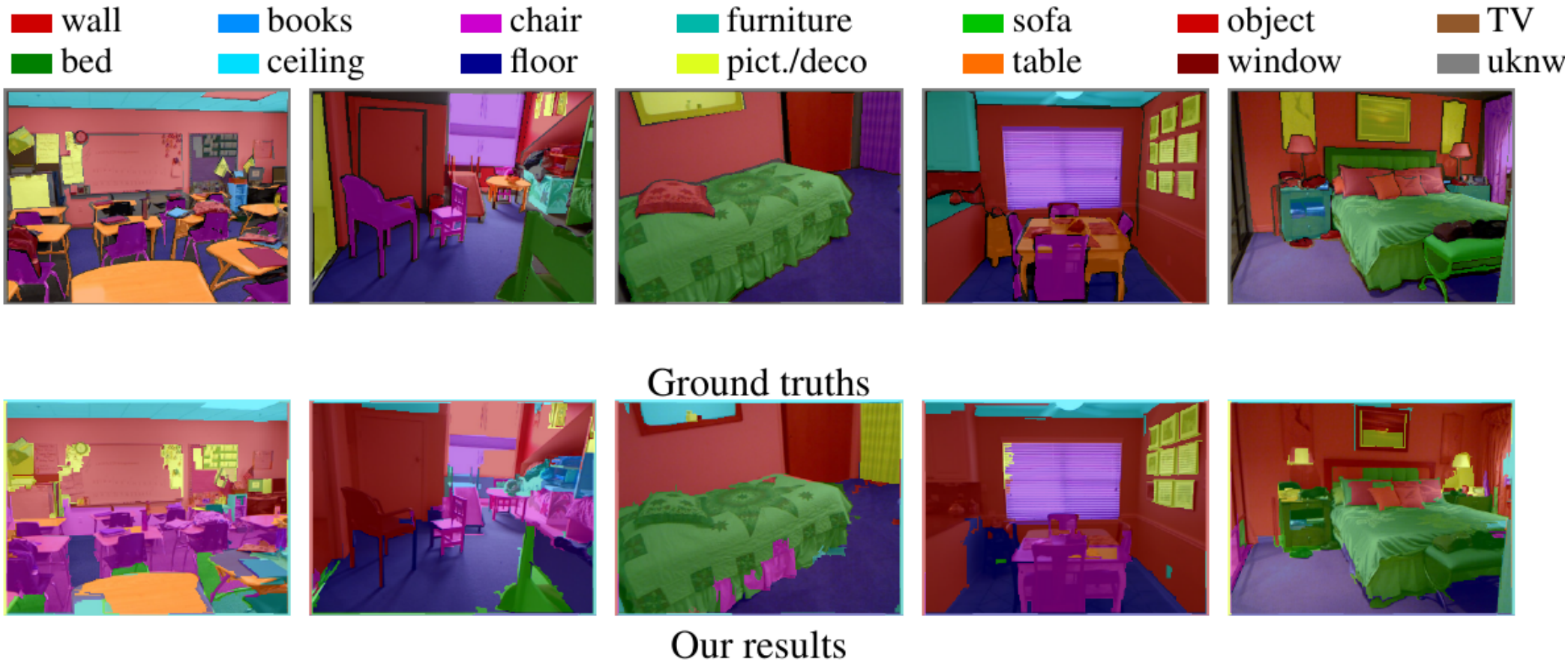
Features from Convolutional net (d=768 per pixel)

Convolutional classifier

"soft" categories scores

[Farabet et al. IEEE T. PAMI 2013]

Scene Parsing/Labeling on RGB+Depth Images



[Couprie, Farabet, Najman, LeCun ICLR 2013, ICIP 2013]

Scene Parsing/Labeling

- No post-processing
- Frame-by-frame
- ConvNet runs at 50ms/frame on Virtex-6 FPGA hardware
  - But communicating the features over ethernet limits system performance

VIDEO: SCENE PARSING

[Farabet et al. ICML 2012, PAMI 2013]

Scene Parsing/Labeling: Performance

- Stanford Background Dataset [Gould 1009]: 8 categories

	Pixel Acc.	Class Acc.	CT (sec.)
Gould *et al.* 2009 [14]	76.4%	-	10 to 600s
Munoz *et al.* 2010 [32]	76.9%	66.2%	12s
Tighe *et al.* 2010 [46]	77.5%	-	10 to 300s
Socher *et al.* 2011 [45]	78.1%	-	?
Kumar *et al.* 2010 [22]	79.4%	-	$< 600s$
Lempitzky *et al.* 2011 [28]	**81.9%**	72.4%	$> 60s$
singlescale convnet	66.0 %	56.5 %	0.35s
multiscale convnet	78.8 %	72.4%	0.6s
multiscale net + superpixels	80.4%	74.56%	**0.7s**
multiscale net + gPb + cover	80.4%	75.24%	61s
multiscale net + CRF on gPb	81.4%	**76.0%**	60.5s

[Rejected from CVPR 2012]

[Farabet et al. ICML 2012] [Farabet et al. IEEE T. PAMI 2013]

Scene Parsing/Labeling: Performance

	Pixel Acc.	Class Acc.
Liu *et al.* 2009 [31]	74.75%	-
Tighe *et al.* 2010 [44]	76.9%	29.4%
raw multiscale net[1]	67.9%	45.9%
multiscale net + superpixels[1]	71.9%	**50.8%**
multiscale net + cover[1]	72.3%	**50.8%**
multiscale net + cover[2]	**78.5%**	29.6%

- SIFT Flow Dataset
- [Liu 2009]:
- 33 categories

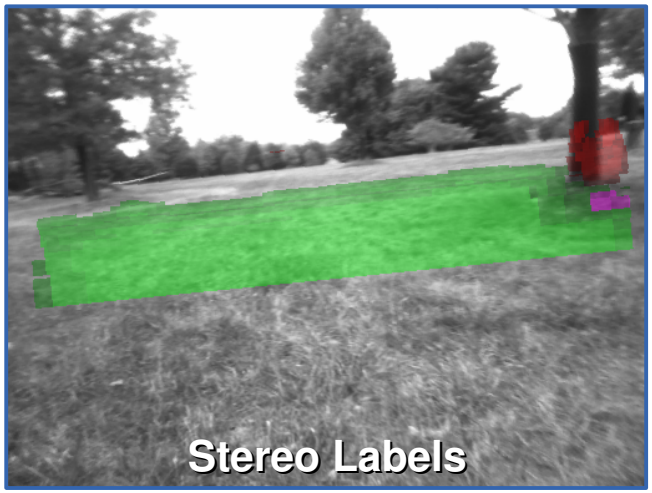
- Barcelona dataset
- [Tighe 2010]:
- 170 categories.

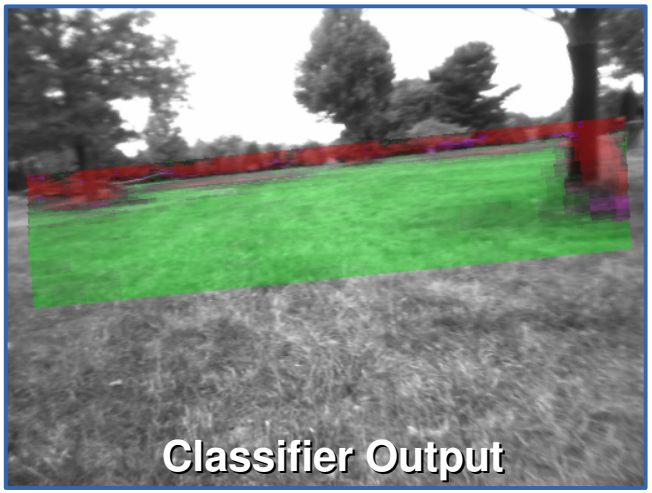
	Pixel Acc.	Class Acc.
Tighe *et al.* 2010 [44]	66.9%	7.6%
raw multiscale net[1]	37.8%	**12.1%**
multiscale net + superpixels[1]	44.1%	**12.4%**
multiscale net + cover[1]	46.4%	**12.5%**
multiscale net + cover[2]	**67.8%**	**9.5%**

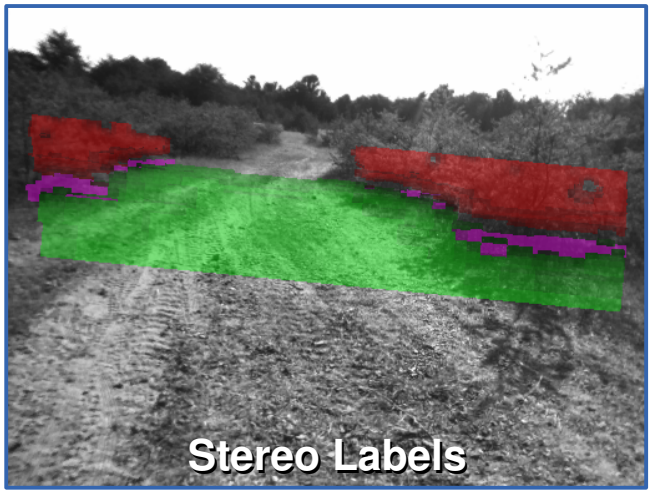
[Farabet et al. IEEE T. PAMI 2012]

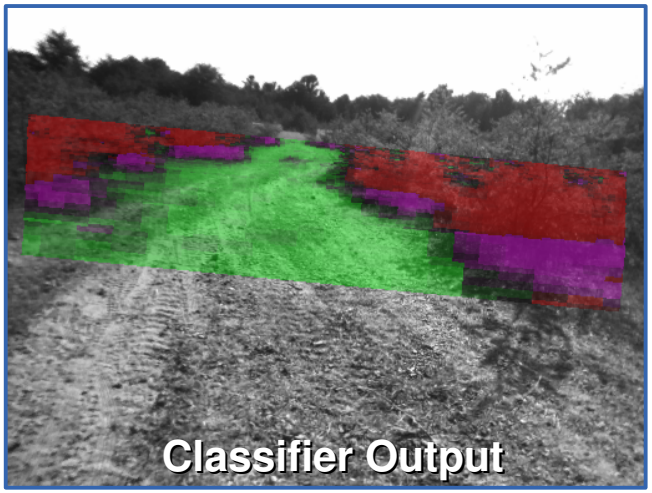
ConvNet for Long Range Adaptive Robot Vision (DARPA LAGR program 2005-2008)

Y LeCun



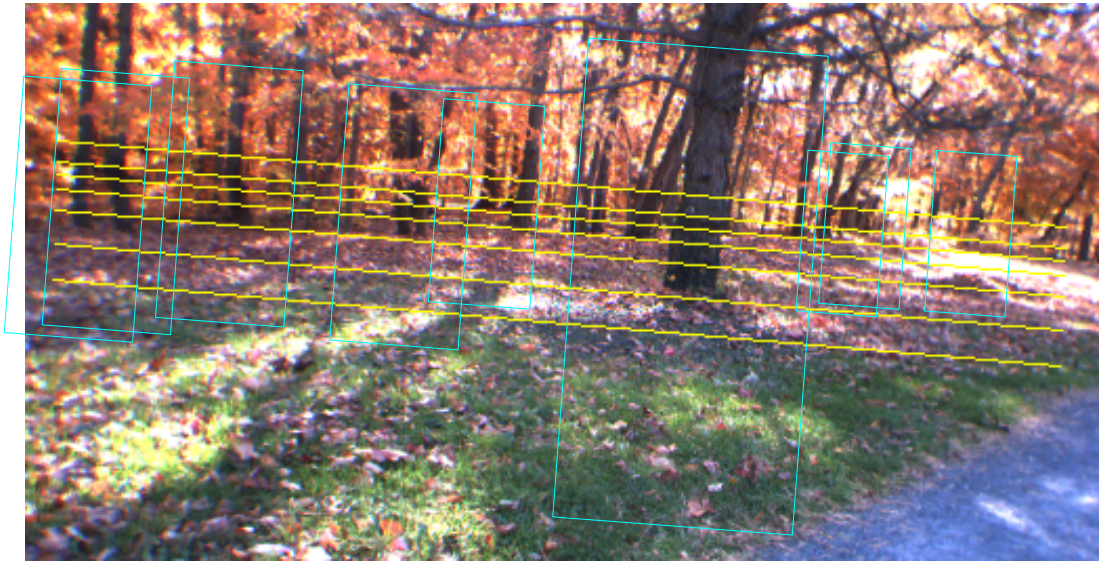






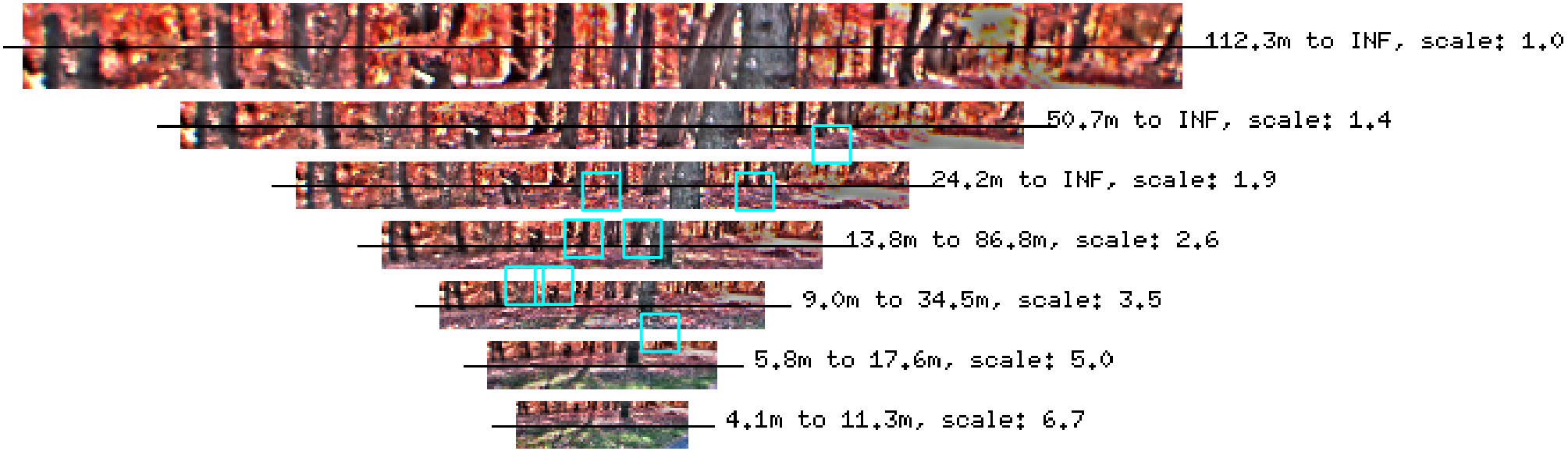
Long Range Vision with a Convolutional Net

Y LeCun



Pre-processing (125 ms)

- Ground plane estimation
- Horizon leveling
- Conversion to YUV + local contrast normalization
- Scale invariant pyramid of distance-normalized image "bands"



Convolutional Net Architecture

Y LeCun

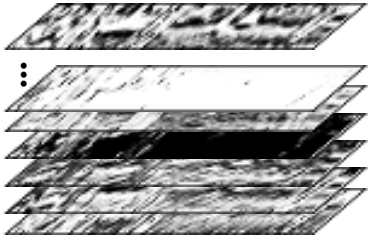
100 features per
3x12x25 input window

100@25x121

CONVOLUTIONS (6x5)

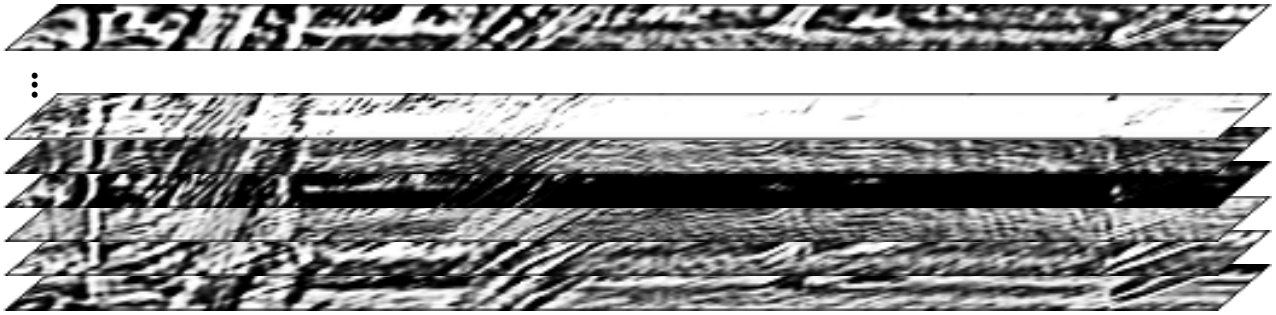
VIDEO: LAGR

20@30x125



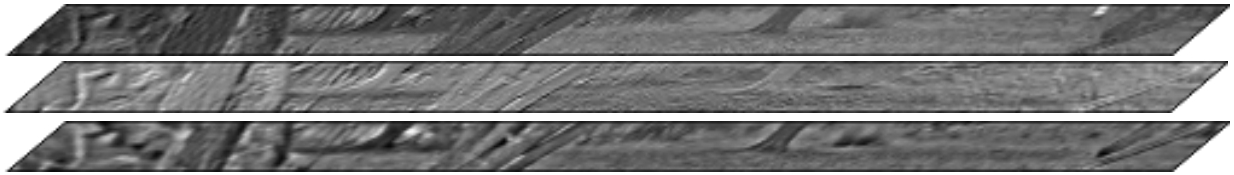
MAX SUBSAMPLING (1x4)

20@30x484



CONVOLUTIONS (7x6)

3@36x484



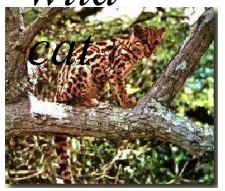
YUV image band
20-36 pixels tall,
36-500 pixels wide

YUV input

- In the mid 2000s, ConvNets were getting decent results on object classification
- Dataset: "Caltech101":
  - 101 categories
  - 30 training samples per category
- But the results were slightly worse than more "traditional" computer vision methods, because:
  - 1. the datasets were too small
  - 2. the computers were too slow

face


beaver


wild cat


lotus


an


dollar


metronome


w. chair


minar


cellphon


joshua t.


cougar body


backgroun d


Then., two things happened...

- **The ImageNet dataset [Fei-Fei et al. 2012]**
  - 1.2 million training samples
  - 1000 categories

- **Fast & Programmable General-Purpose GPUs**
  - NVIDIA CUDA
  - Capable of over 1 trillion operations/second

Matchstick

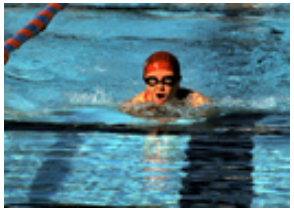
Flute

Backpack

Sea lion

Strawberry

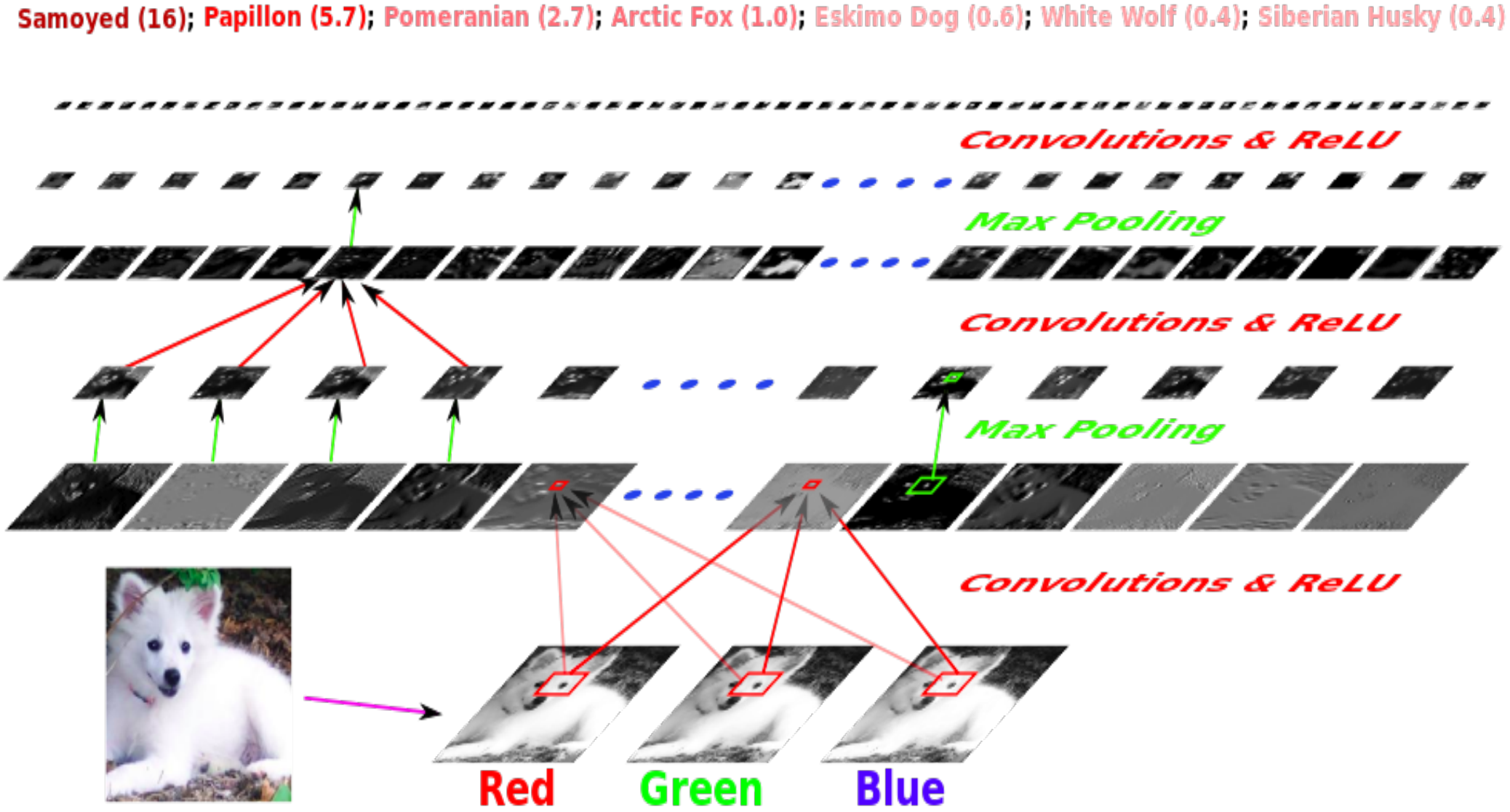
Bathing cap



Racket

Very Deep ConvNet for Object Recognition

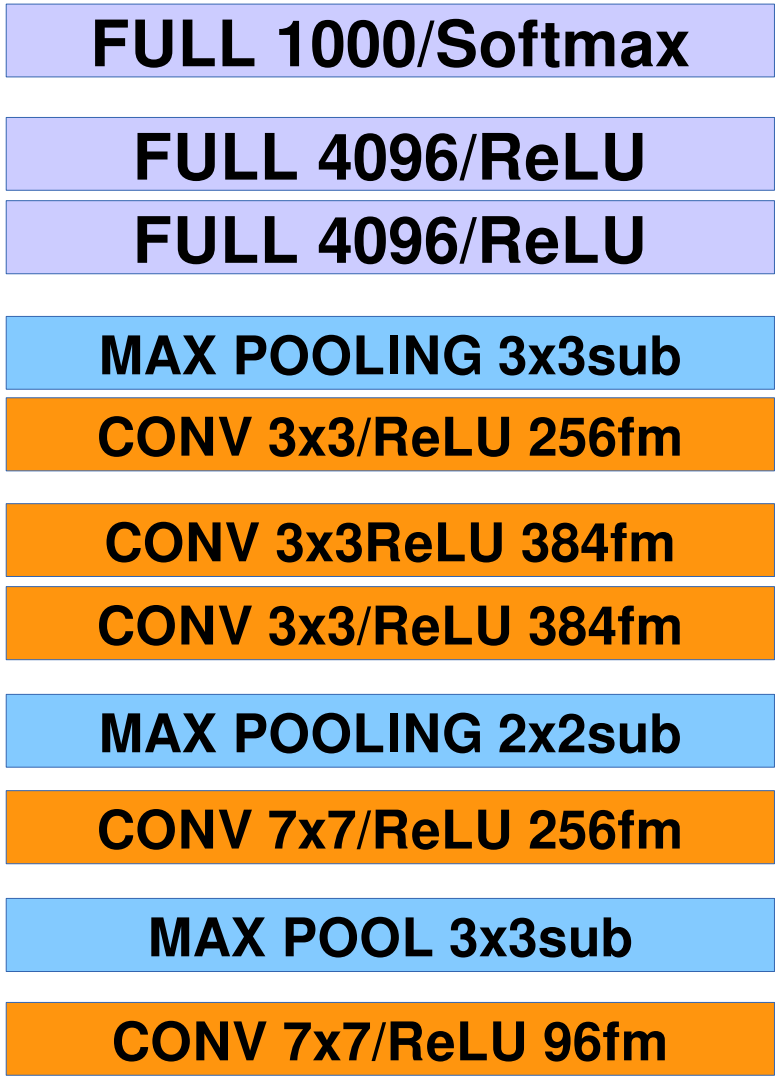
- 1 to 10 billion connections, 10 million to 1 billion parameters, 8 to 20 layers.



Very Deep ConvNets Trained on GPU

- AlexNet [Krizhevski, Sutskever, Hinton 2012]
  - 15% top-5 error on ImageNet
- OverFeat [Sermanet et al. 2013]
  - 13.8%
- VGG Net [Simonyan, Zisserman 2014]
  - 7.3%
- GoogLeNet [Szegedy et al. 2014]
  - 6.6%
- ResNet [He et al. 2015]
  - 5.7%

- http://torch.ch
- https://github.com/torch/torch7/wiki/Cheatsheet

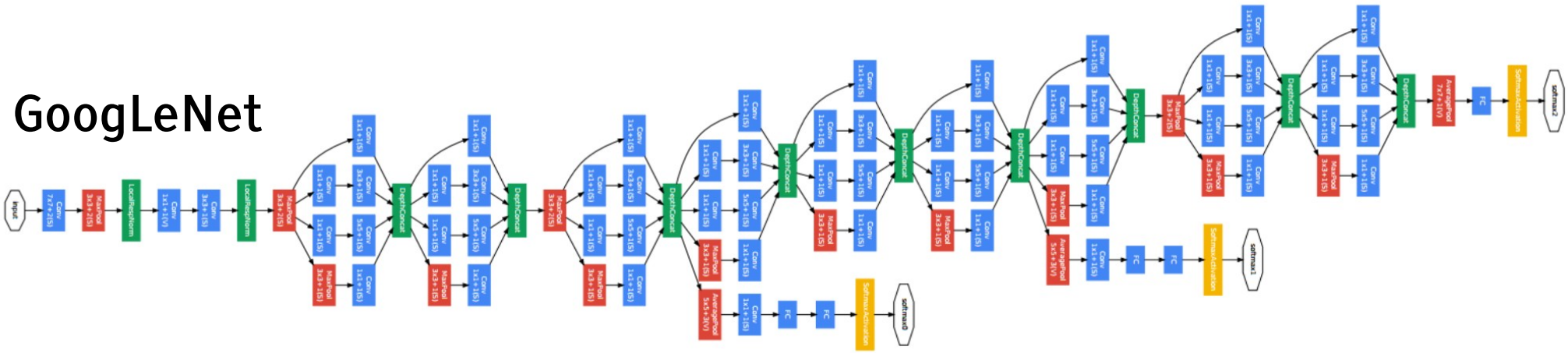


Very Deep ConvNet Architectures

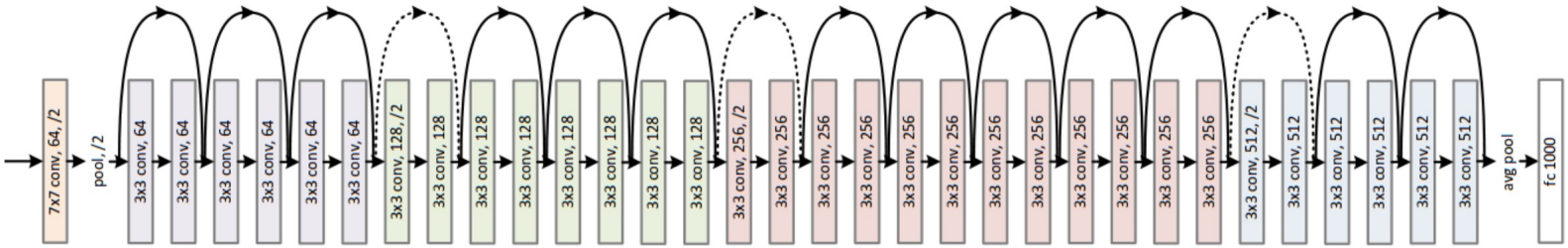
- Small kernels, not much subsampling (fractional subsampling).

VGG

GoogLeNet

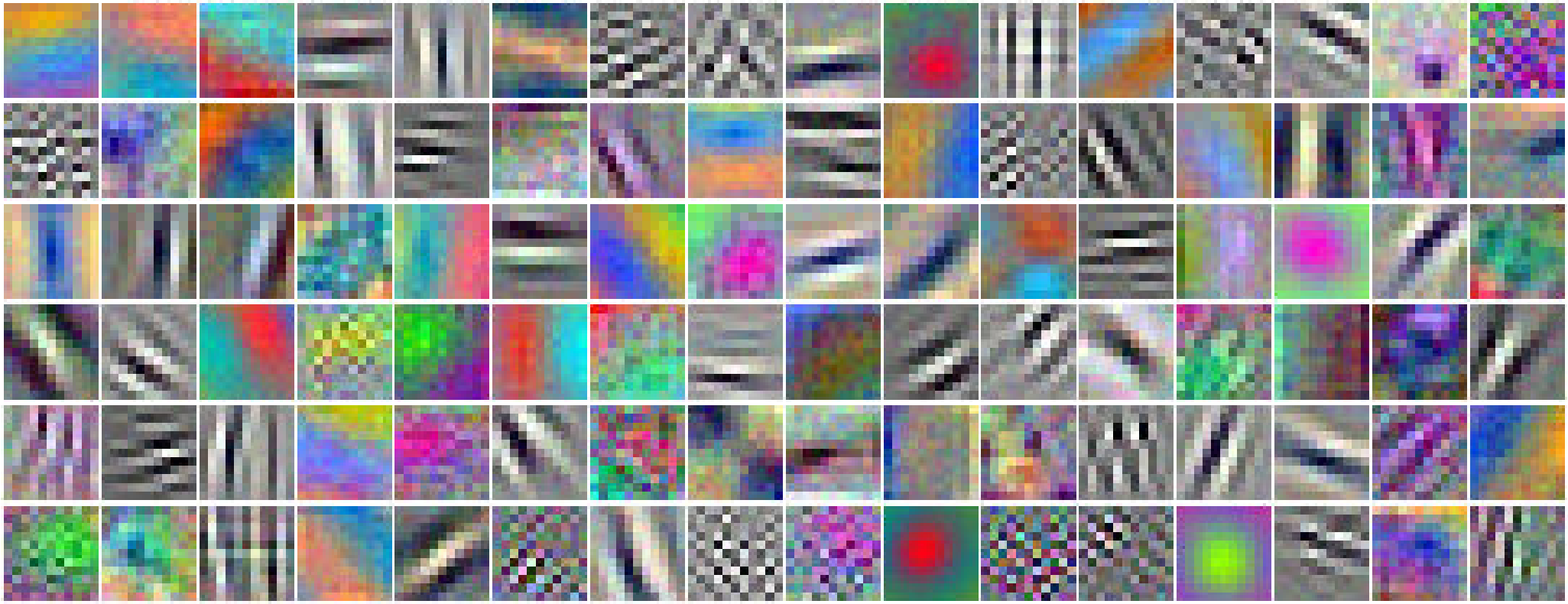


ResNet



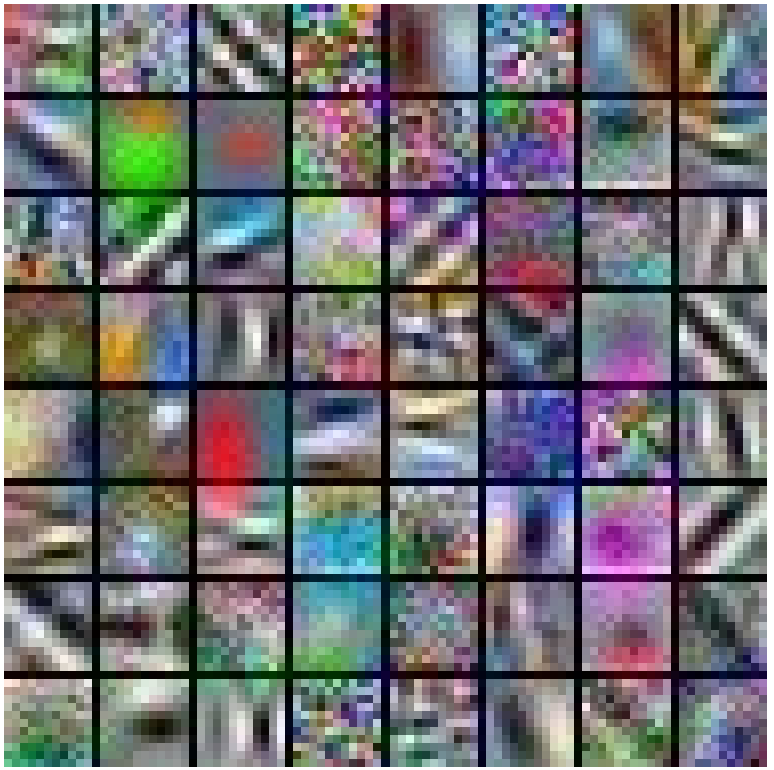
Kernels: Layer 1 (11x11)

- Layer 1: 3x96 kernels, RGB->96 feature maps, 11x11 Kernels, stride 4



Learning in Action

- **How the filters in the first layer learn**

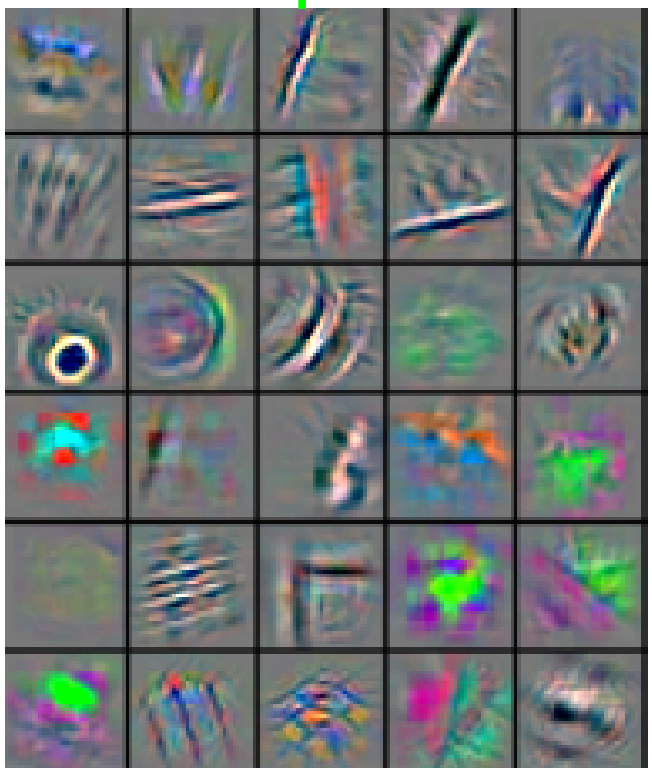


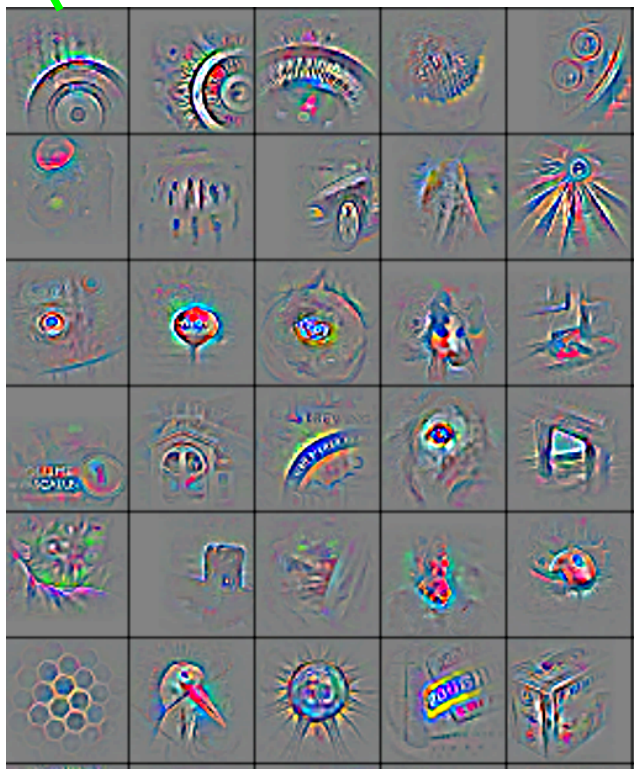
Deep Learning = Learning Hierarchical Representations

Y LeCun

- It's deep if it has more than one stage of non-linear feature transformation

Low-Level Feature → Mid-Level Feature → High-Level Feature → Trainable Classifier





Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

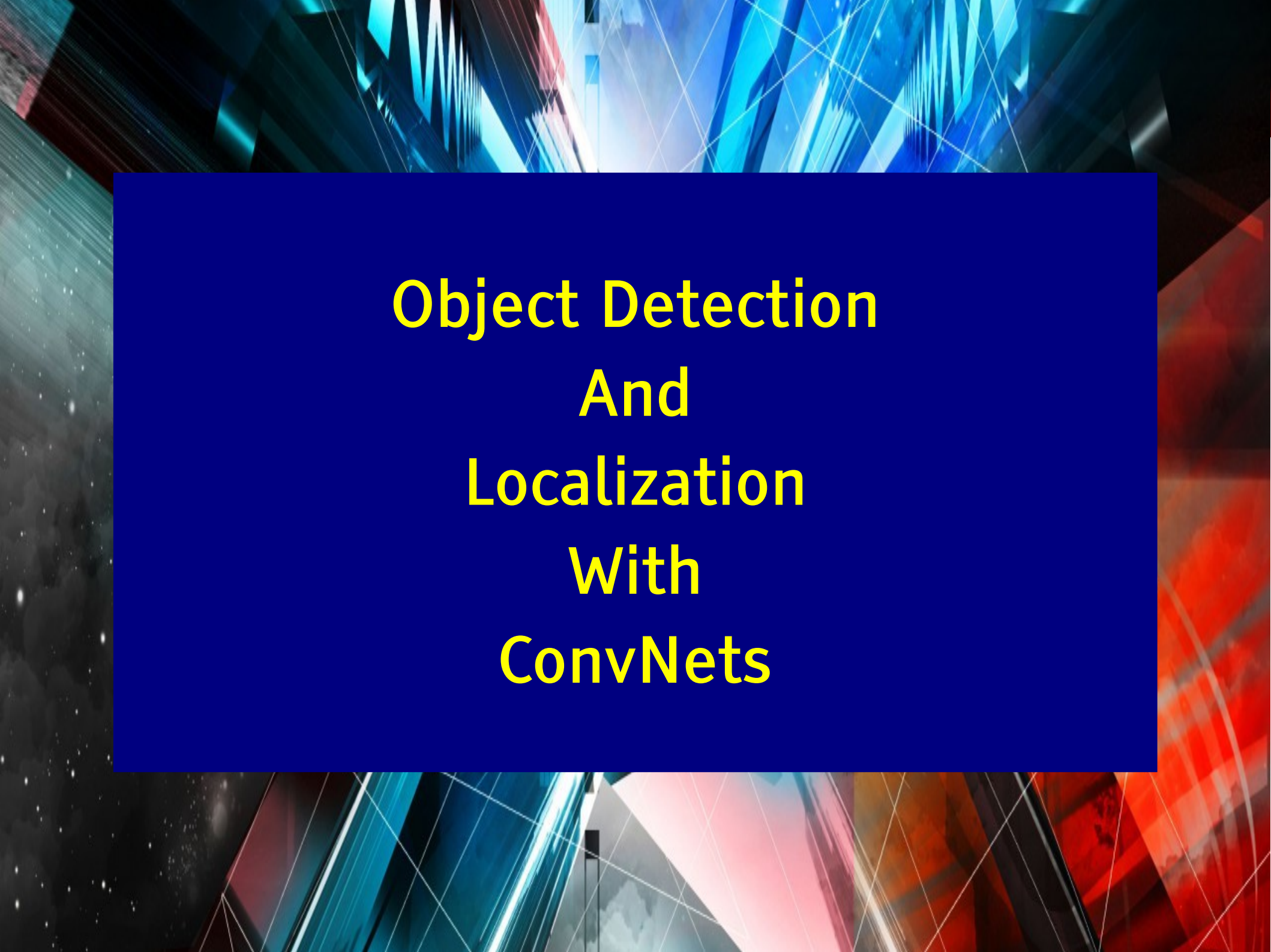
ImageNet: Classification

- Give the name of the dominant object in the image
- Top-5 error rates: if correct class is not in top 5, count as error
  - Red:ConvNet, blue: no ConvNet

2012 Teams	%error
Supervision (Toronto)	15.3
ISI (Tokyo)	26.1
VGG (Oxford)	26.9
XRCE/INRIA	27.0
UvA (Amsterdam)	29.6
INRIA/LEAR	33.4

2013 Teams	%error
Clarifai (NYU spinoff)	11.7
NUS (singapore)	12.9
Zeiler-Fergus (NYU)	13.5
A. Howard	13.5
OverFeat (NYU)	14.1
UvA (Amsterdam)	14.2
Adobe	15.2
VGG (Oxford)	15.2
VGG (Oxford)	23.0

2014 Teams	%error
GoogLeNet	6.6
VGG (Oxford)	7.3
MSRA	8.0
A. Howard	8.1
DeeperVision	9.5
NUS-BST	9.7
TTIC-ECP	10.2
XYZ	11.2
UvA	12.1


Object Detection
And
Localization
With
ConvNets

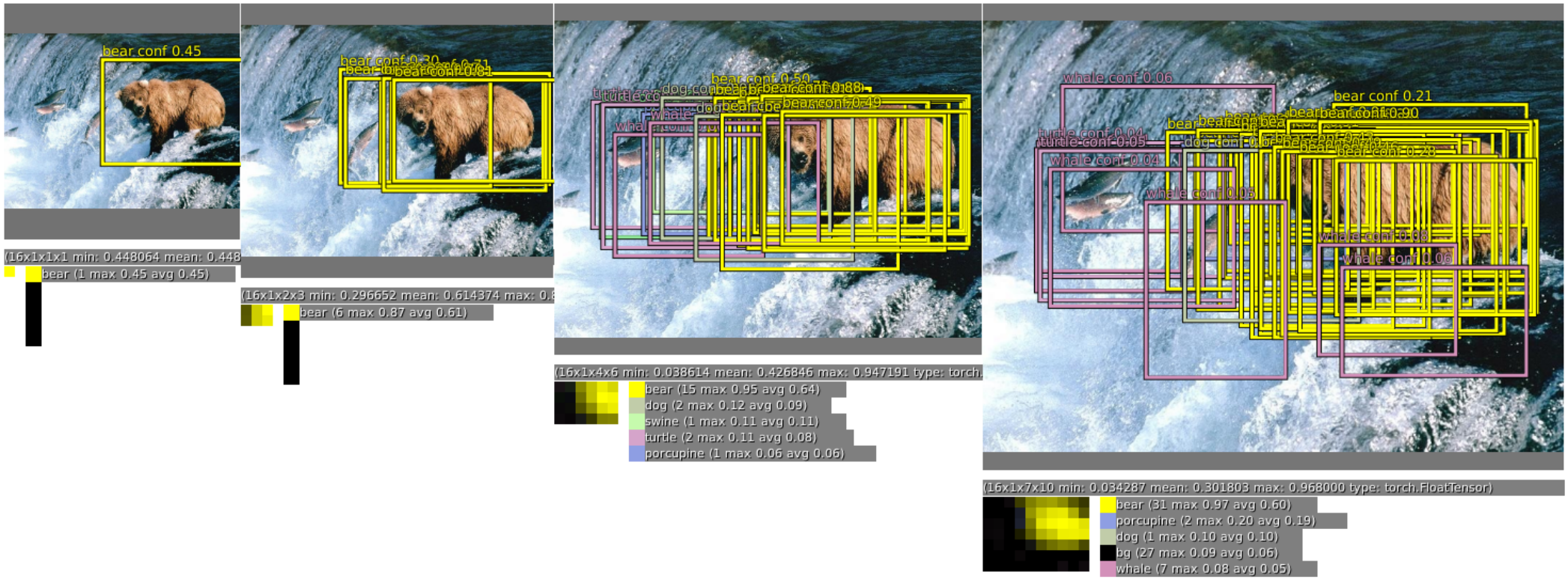
Classification + Localization: multiscale sliding window

- Apply convnet with a sliding window over the image at multiple scales
- Important note: it's very cheap to slide a convnet over an image
  - Just compute the convolutions over the whole image and replicate the fully-connected layers

Classification + Localization: sliding window + bounding box regression

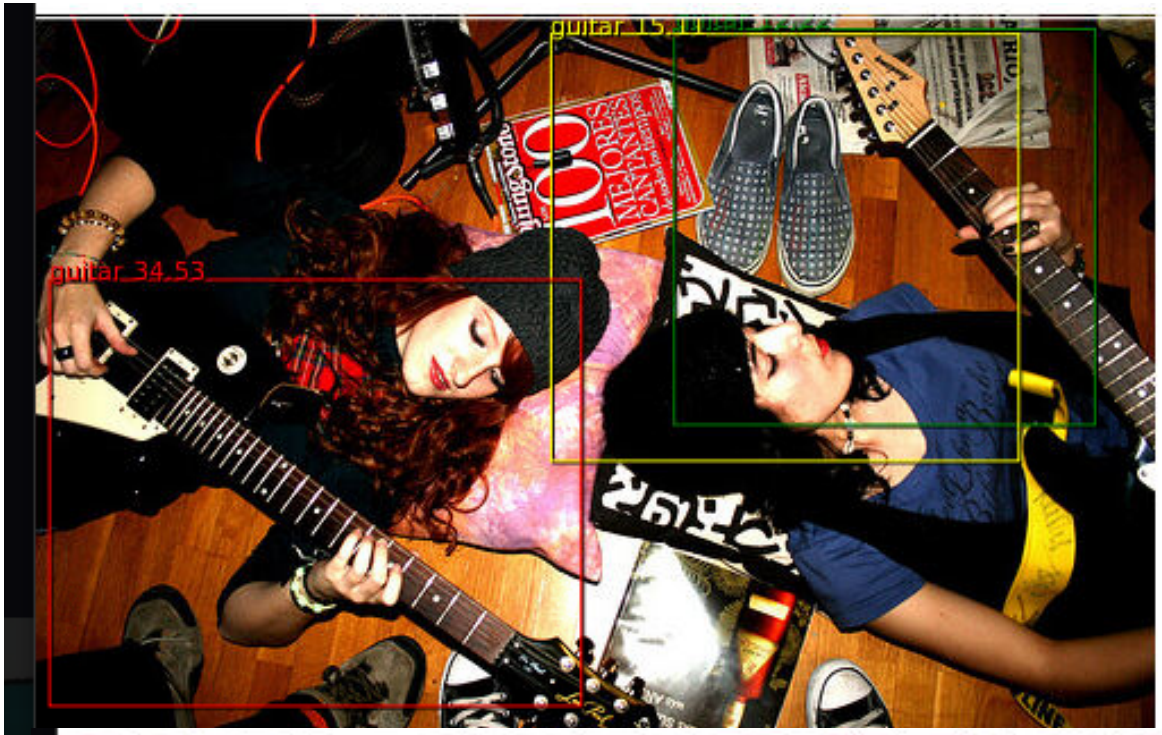
Y LeCun

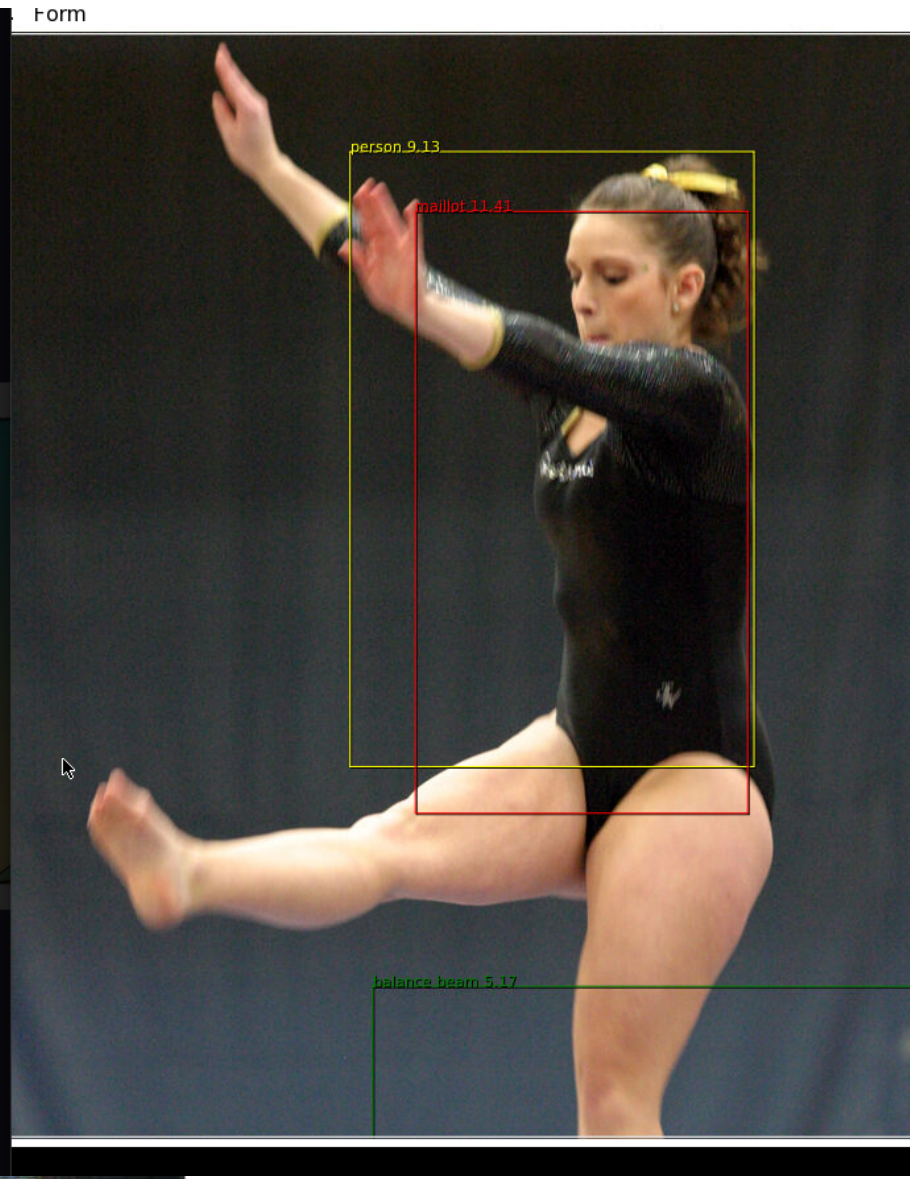
- Apply convnet with a sliding window over the image at multiple scales
- For each window, predict a class and bounding box parameters
  - Evenif the object is not completely contained in the viewing window, the convnet can predict where it thinks the object is.

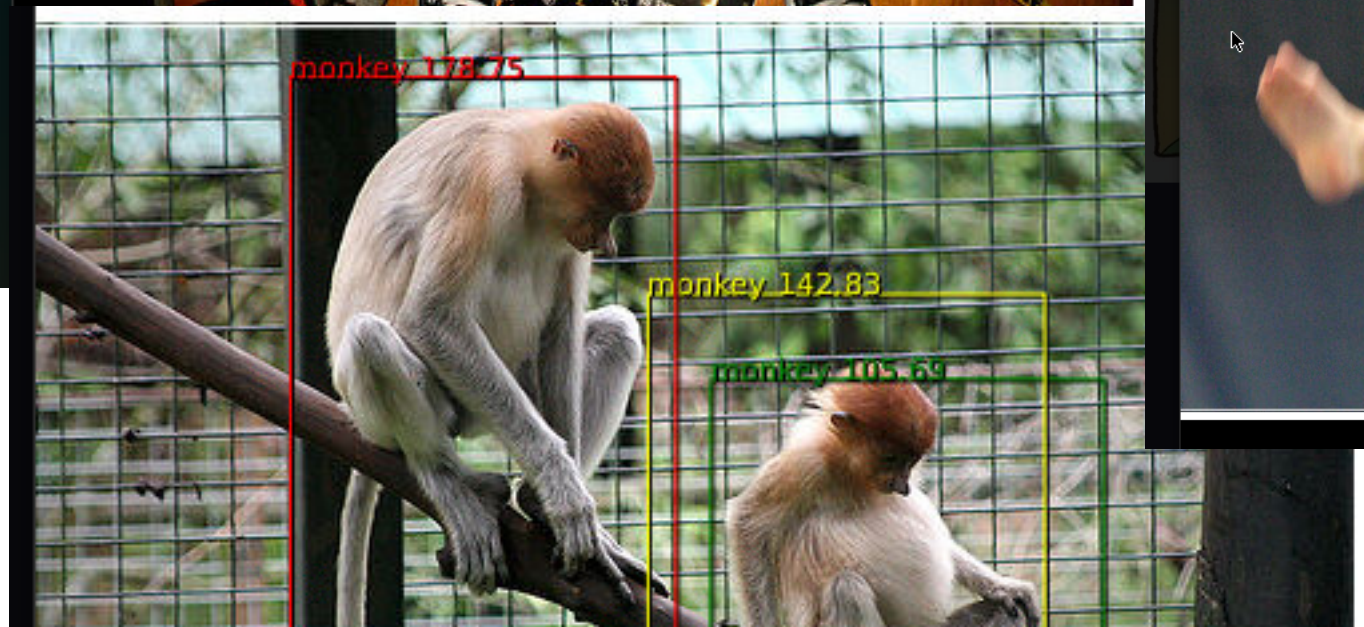


Results: pre-trained on ImageNet1K, fine-tuned on ImageNet Detection

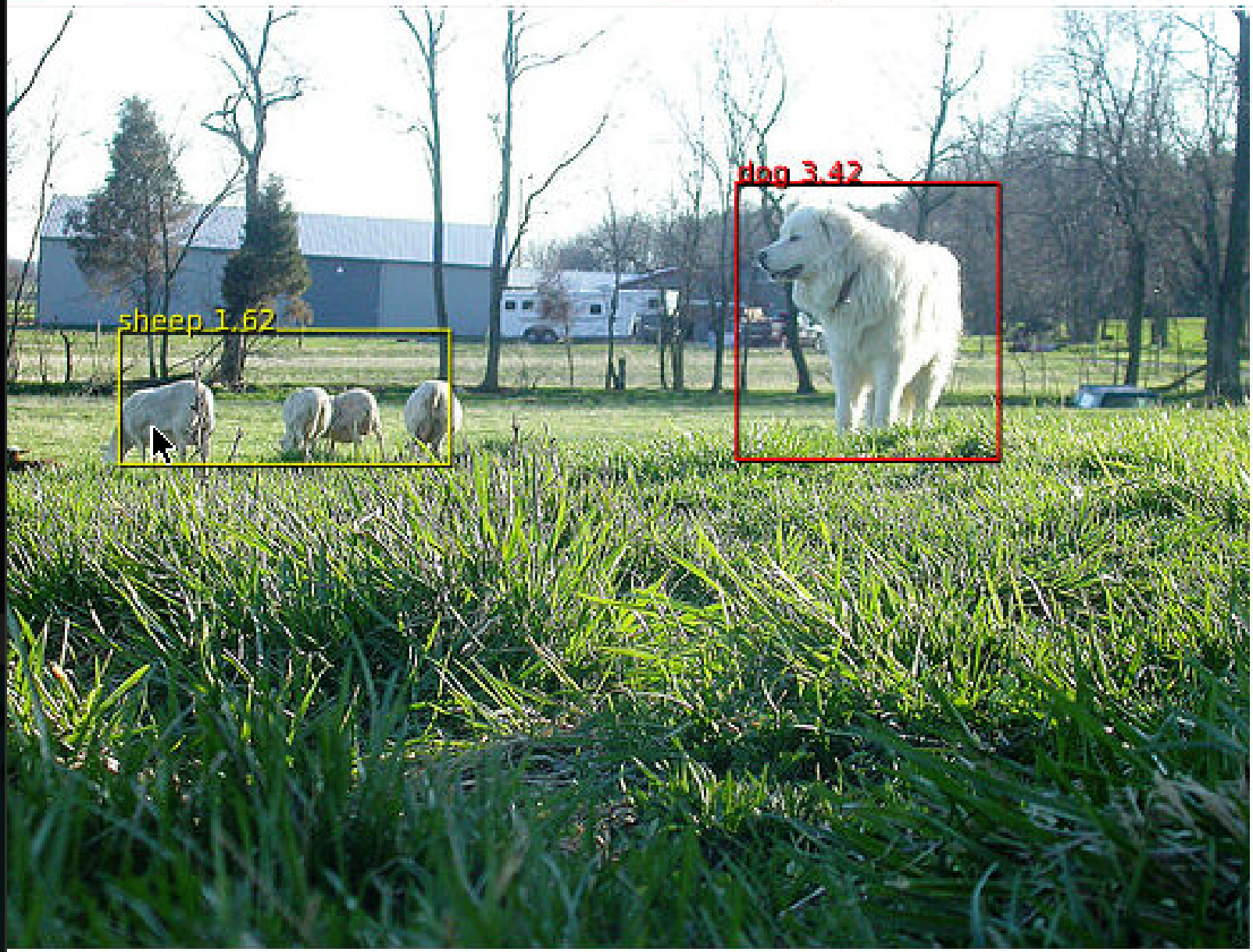
Y LeCun





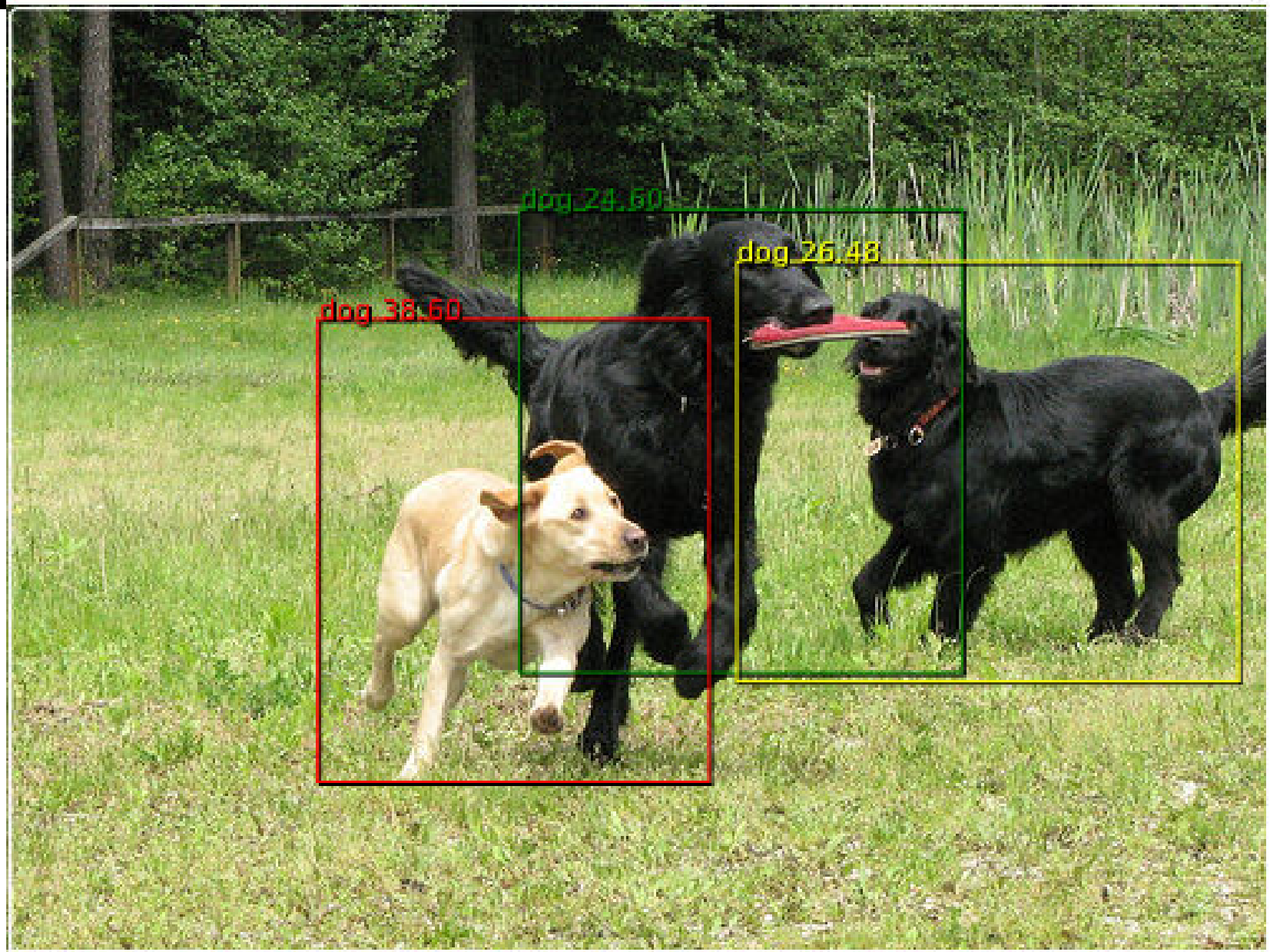


Detection Examples



/home/snwiz/data/imagenet12/original/det/ILSVRC2013_DET_test/ILSVRC2012_test_00090628.JPEG
dog conf 3.419652

Detection Examples

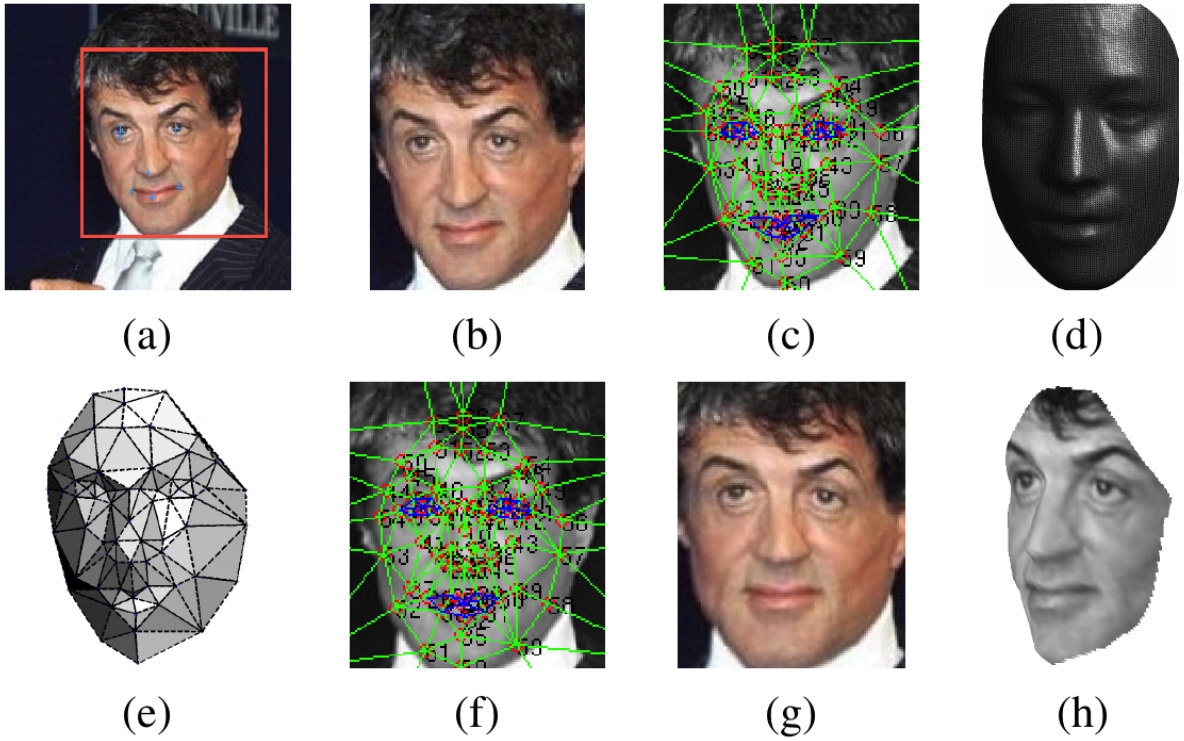


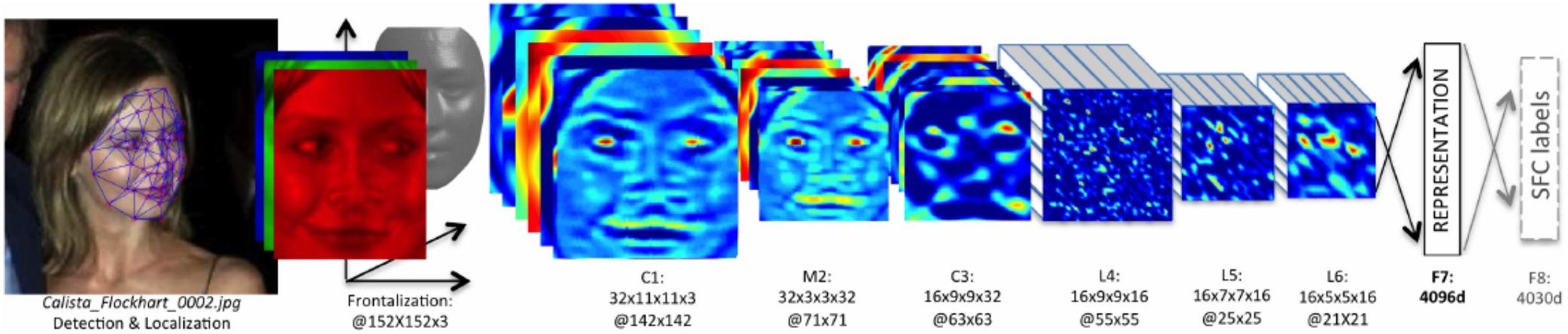
/home/snwiz/data/imagenet12/original/det/ILSVRC2013_DET_test/ILSVRC2012_test_00000172.JPEG
dog conf 38.603936

Detection Examples

Deep Face

- **[Taigman et al. CVPR 2014]**
  - Alignment
  - ConvNet
  - Metric Learning
- **Deployed at Facebook for Auto-tagging**
  - 800 million photos per day

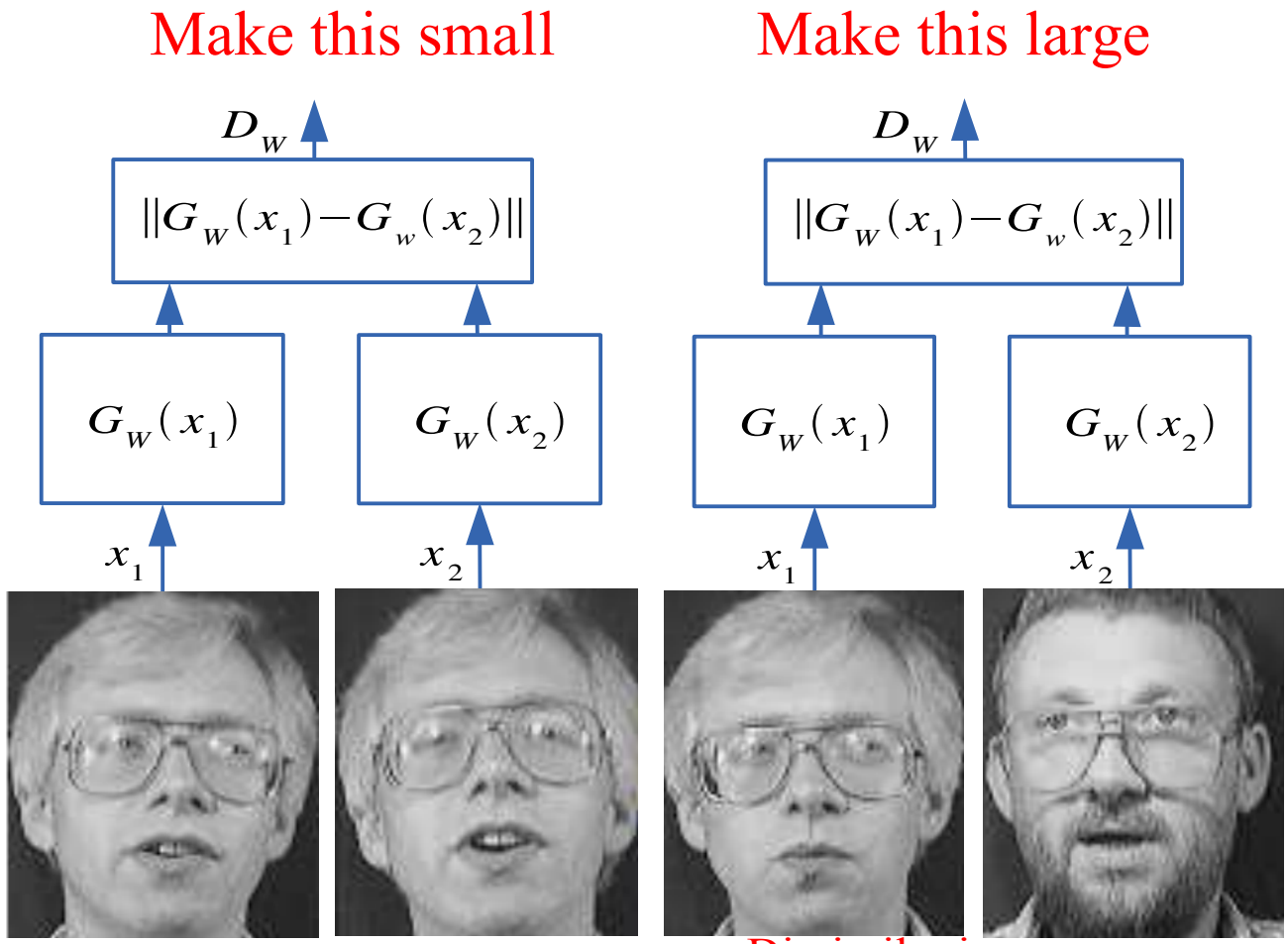




Metric Learning with a Siamese Architecture

Contrative Obective Function

- Similar objects should produce outputs that are nearby
- Dissimilar objects should produce output that are far apart.
- DrLIM: Dimensionality Reduction by Learning and Invariant Mapping
- [Chopra et al. CVPR 2005]
- [Hadsell et al. CVPR 2006]



Pose Estimation and Attribute Recovery with ConvNets

Y LeCun

Pose-Aligned Network for Deep Attribute Modeling

[Zhang et al. CVPR 2014] (Facebook AI Research)

(a) Highest scoring results for people wearing glasses.

(b) Highest scoring results for people wearing a hat.

Real-time hand pose recovery

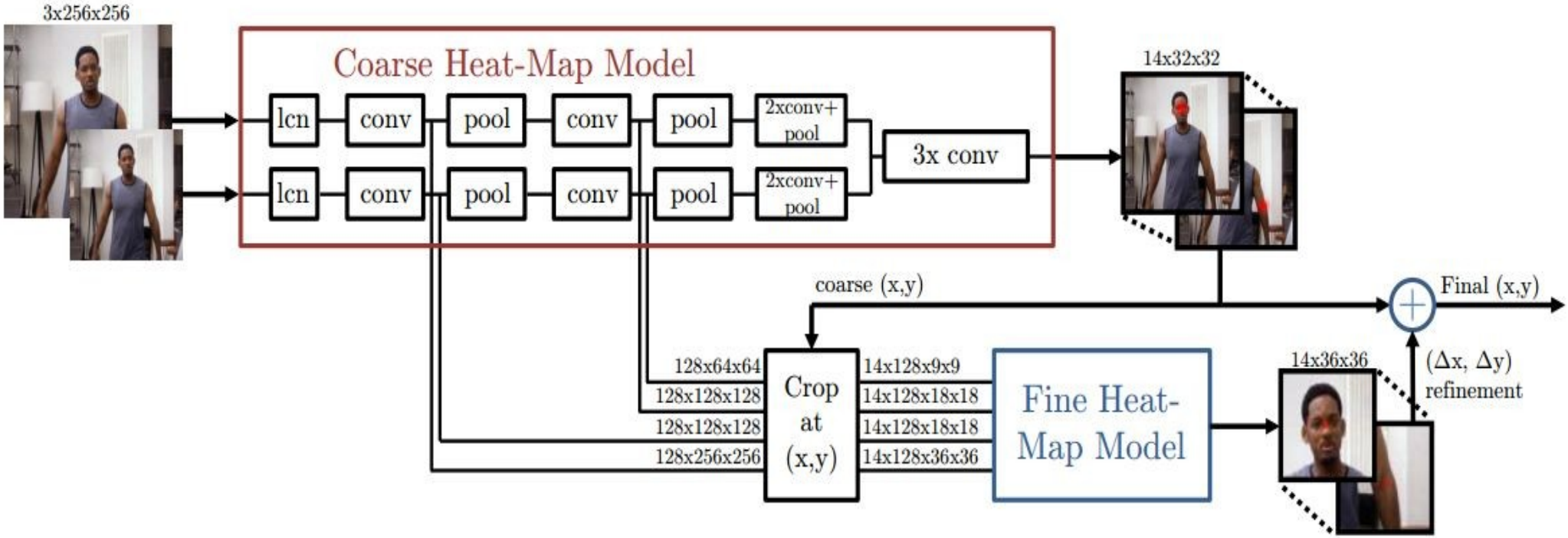
[Tompson et al. Trans. on Graphics 14]

Body pose estimation [Tompson et al. ICLR, 2014]

HAND POSE V IDEO

Person Detection and Pose Estimation

Tompson, Goroshin, Jain, LeCun, Bregler arXiv:1411.4280 (2014)



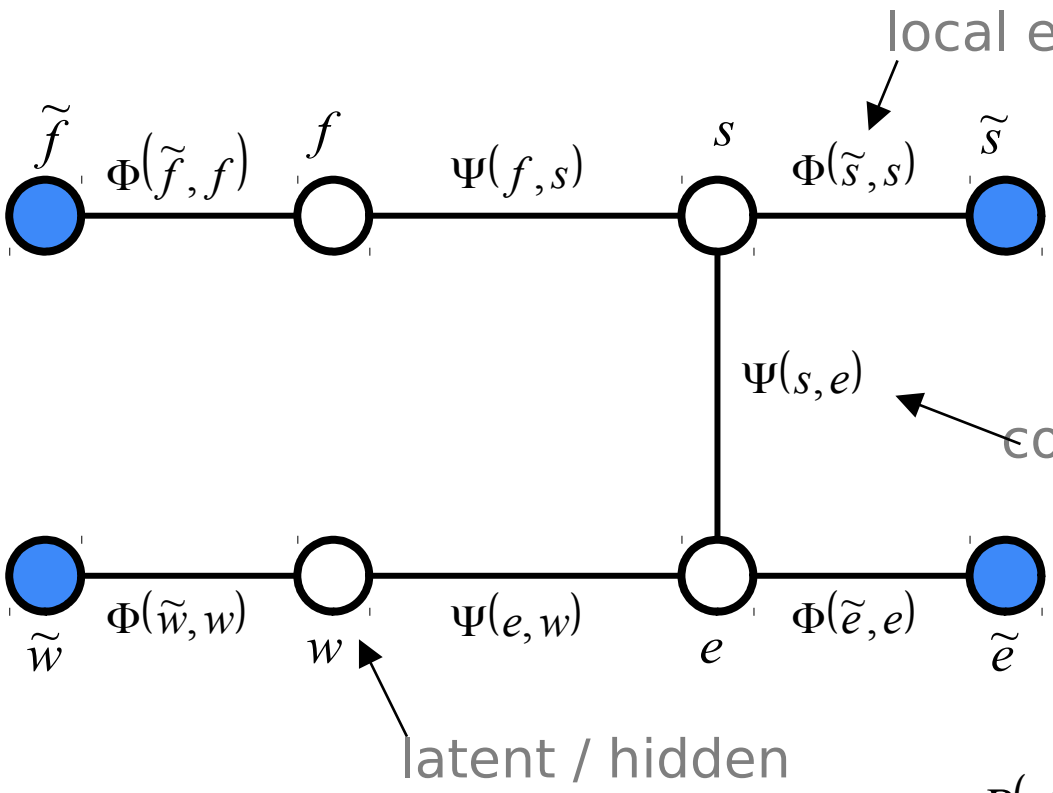
Person Detection and Pose Estimation

Tompson, Goroshin, Jain, LeCun, Bregler arXiv:1411.4280 (2014)

SPATIAL MODEL

Start with a tree graphical model

MRF over spatial locations



Joint Distribution:

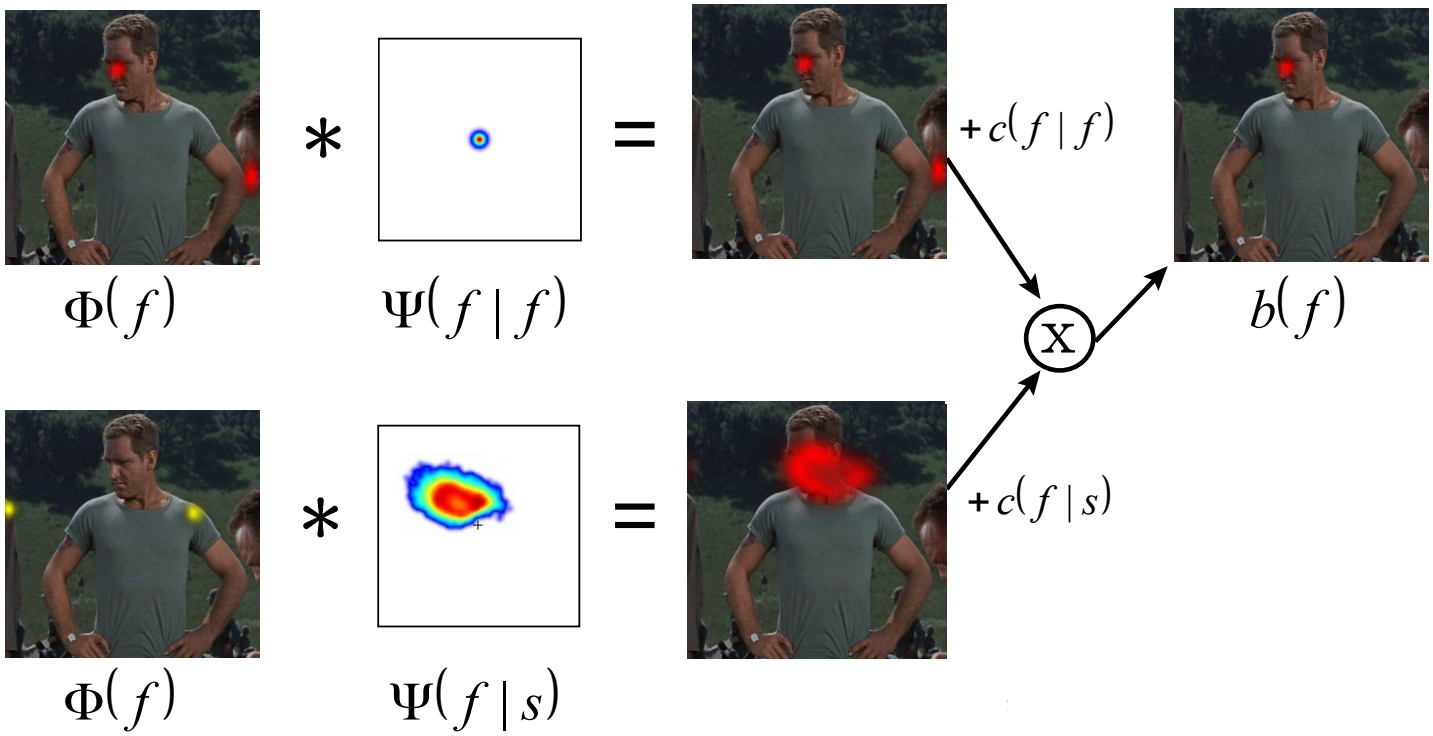
$$P(f,s,e,w) = \frac{1}{Z} \prod_{i,j} \Psi(x_i, x_j) \prod_i \Phi(x_i, \tilde{x}_i)$$

SPATIAL MODEL

Start with a tree graphical model

... And approximate it

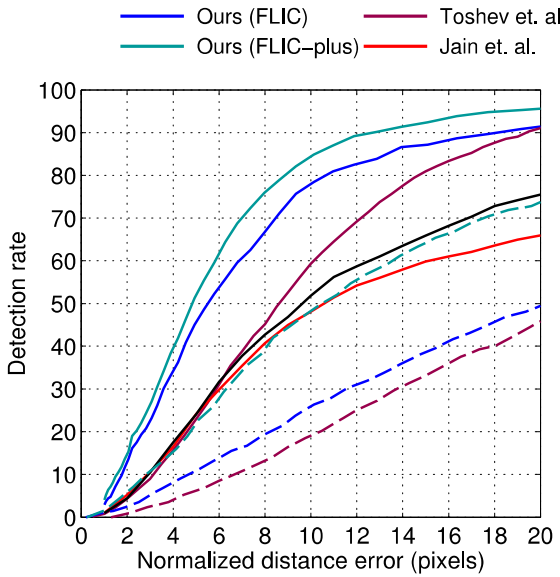
$$b(f) = \Phi(f) \prod_i (\Phi(x_i) \star \Psi(f \mid x_i) + c(f \mid x_i))$$



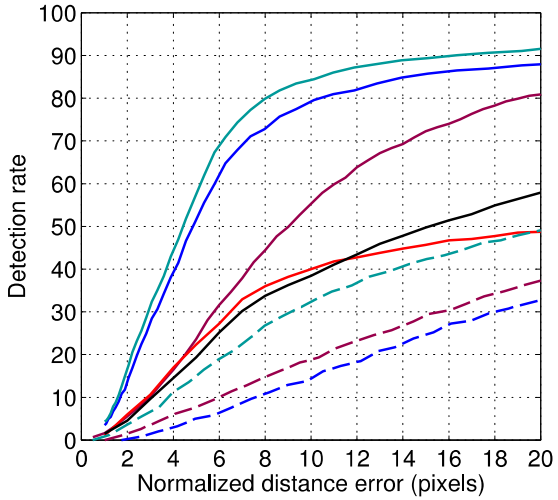
SPATIAL MODEL: RESULTS

Y LeCun

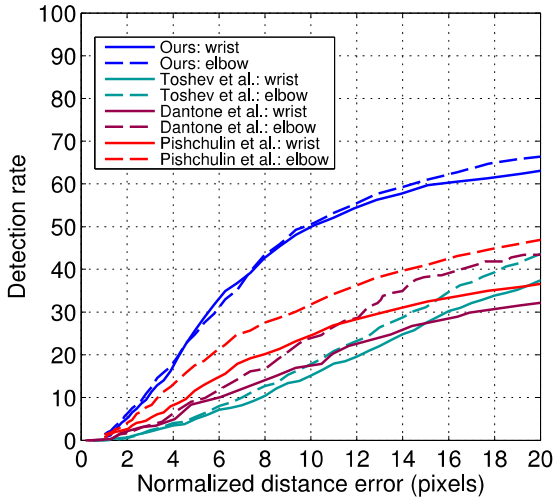
FLIC[1] Elbow



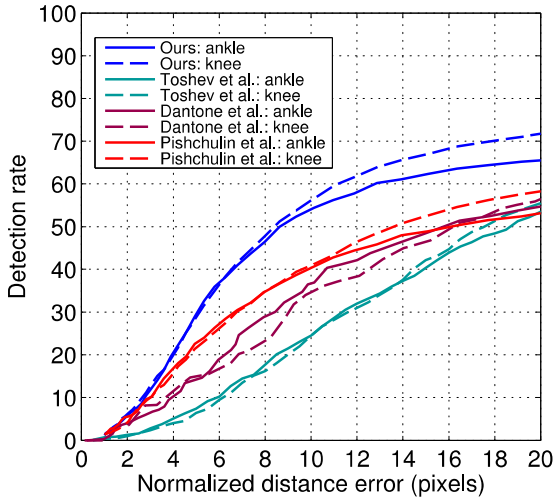
FLIC[1] Wrist



LSP[2] Arms



LSP[1] Legs



(1) B. Sapp and B. Taskar. MODEC: Multimodel decomposition models for human pose estimation. CVPR'13
(2) S. Johnson and M. Everingham. Learning Effective Human Pose Estimation for Inaccurate Annotation. CVPR'11

Image captioning: generating a descriptive sentence

Y LeCun

[Lebret, Pinheiro, Collobert 2015] [Kulkarni 11] [Mitchell 12] [Vinyals 14] [Mao 14]

A man riding skis on a snow covered ski slope.

NP: a man, skis, the snow, a person, a woman, a snow covered slope, a slope, a snowboard, a skier, man.
VP: wearing, riding, holding, standing on, skiing down.
PP: on, in, of, with, down.

A man wearing skis on the snow.

A man is doing skateboard tricks on a ramp.

NP: a skateboard, a man, a trick, his skateboard, the air, a skateboarder, a ramp, a skate board, a person, a woman.
VP: doing, riding, is doing, performing, flying through.
PP: on, of, in, at, with.

A man riding a skateboard on a ramp.

The girl with blue hair stands under the umbrella.

NP: a woman, an umbrella, a man, a person, a girl, umbrellas, that, a little girl, a cell phone.
VP: holding, wearing, is holding, holds, carrying.
PP: with, on, of, in, under.

A woman is holding an umbrella.

A slice of pizza sitting on top of a white plate.

NP: a plate, a white plate, a table, pizza, it, a pizza, food, a sandwich, top, a close.
VP: topped with, has, is, sitting on, is on.
PP: of, on, with, in, up.

A table with a plate of pizza on a white plate.

A baseball player swinging a bat on a field.

NP: the ball, a game, a baseball player, a man, a tennis court, a ball, home plate, a baseball game, a batter, a field.
VP: swinging, to hit, playing, holding, is swinging.
PP: on, during, in, at, of.

A baseball player swinging a bat on a baseball field.

A bunch of kites flying in the sky on the beach.

NP: the beach, a beach, a kite, kites, the ocean, the water, the sky, people, a sandy beach, a group.
VP: flying, flies, is flying, flying in, are.
PP: on, of, with, in, at.

People flying kites on the beach.

C3D: Video Classification with 3D ConvNet

[Tran et al. 2015]

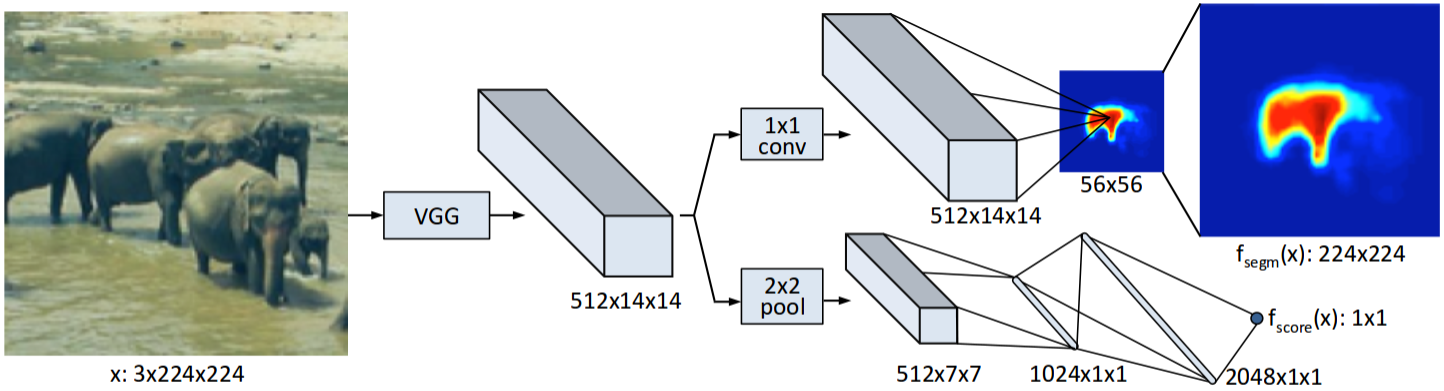
VIDEO: COMMON SPORTS

VIDEO: UNCOMMON SPORTS

Segmenting and Localizing Objects (DeepMask)

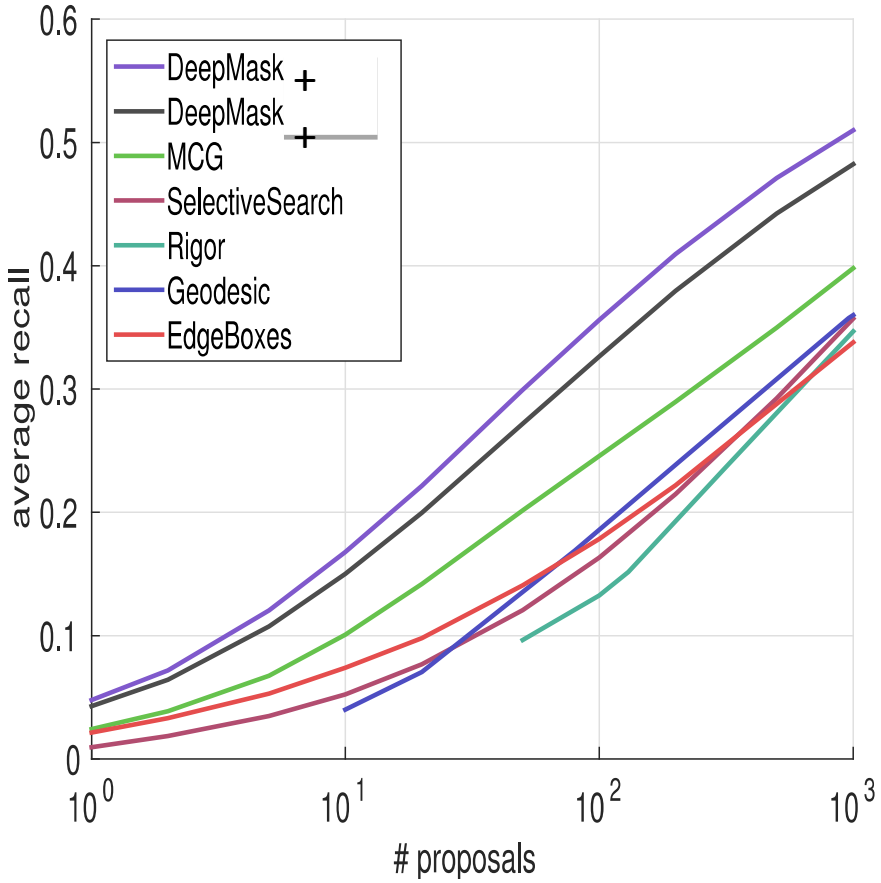
Y LeCun

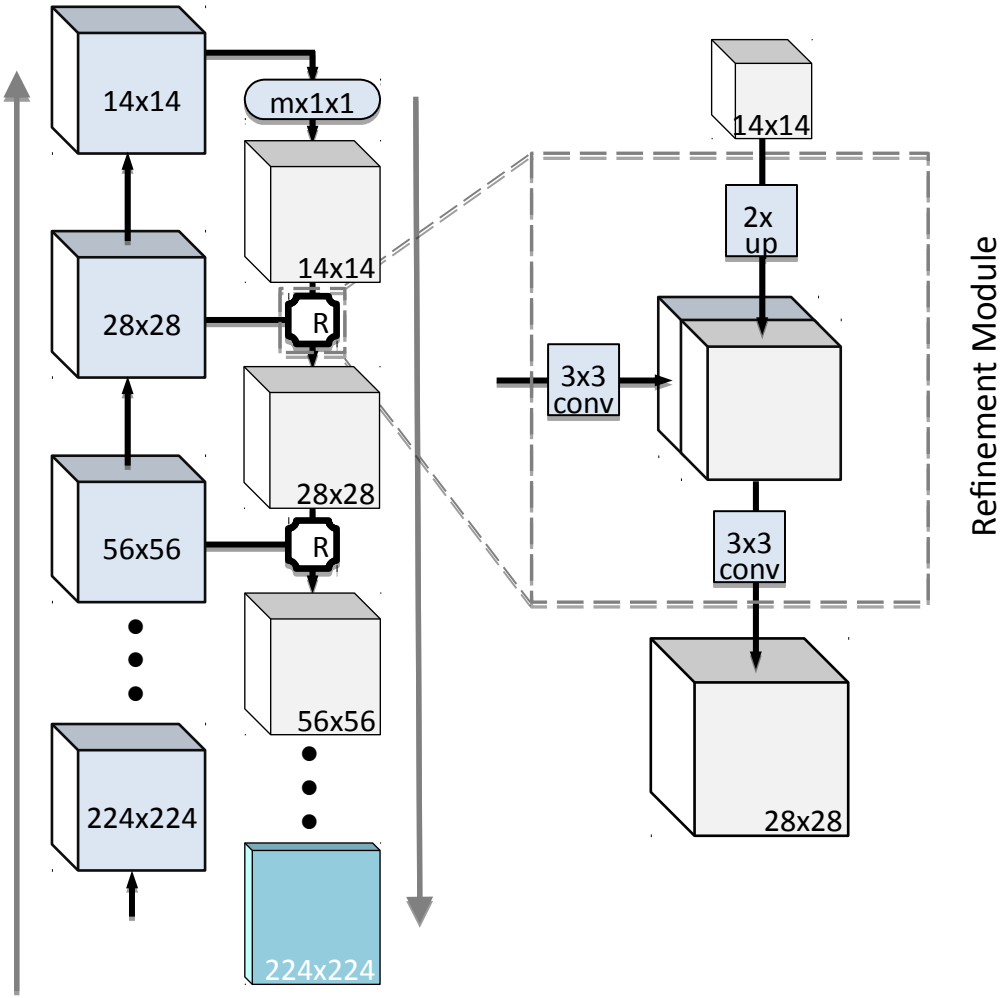
- [Pinheiro, Collobert, Dollar ICCV 2015]
  - ConvNet produces object masks



Iterative Refinement: DeepMask++

FAIR COCO Team



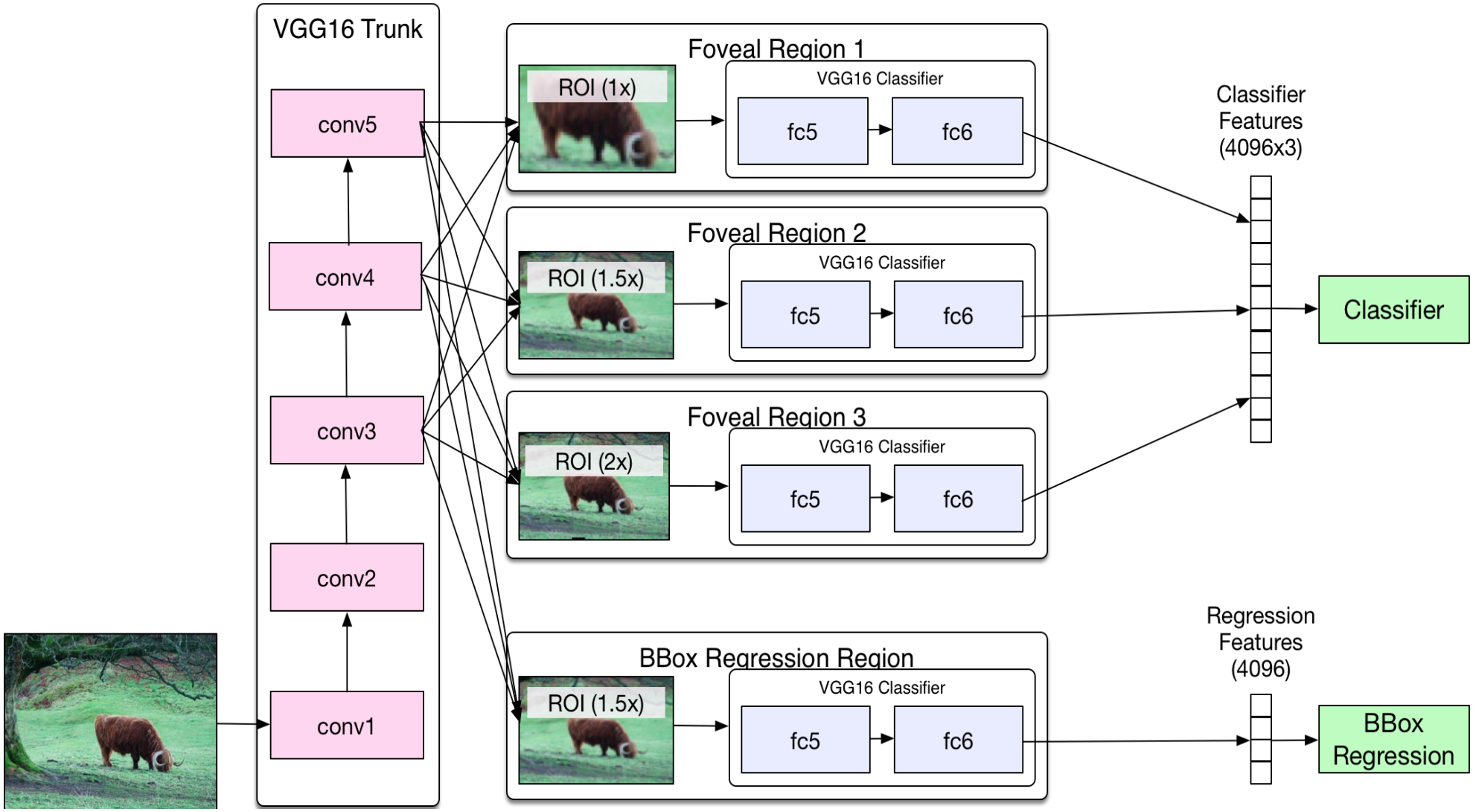


DeepMask++ Proposals

FAIR COCO Team

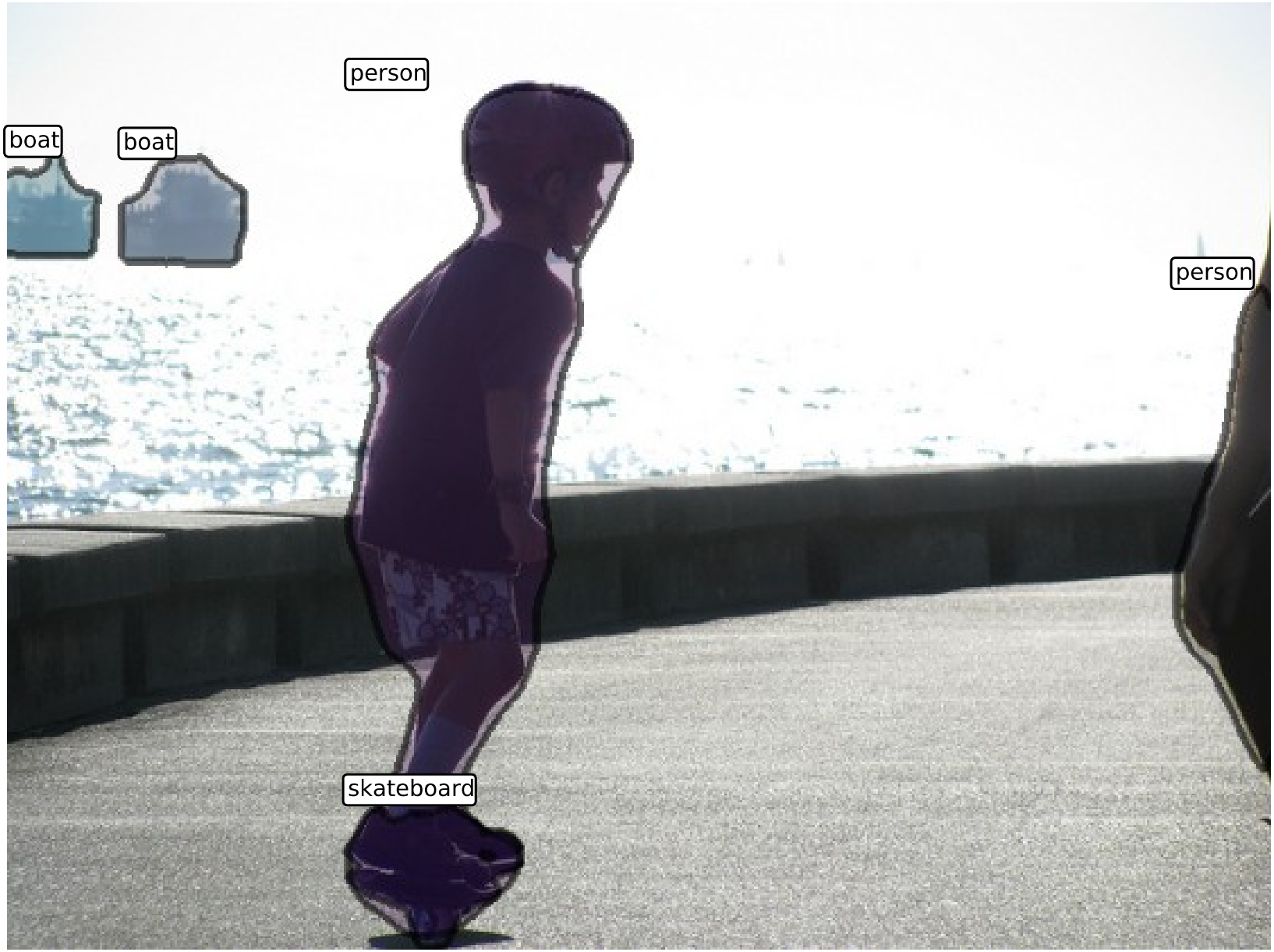
Recognition Pipeline

FAIR COCO Team



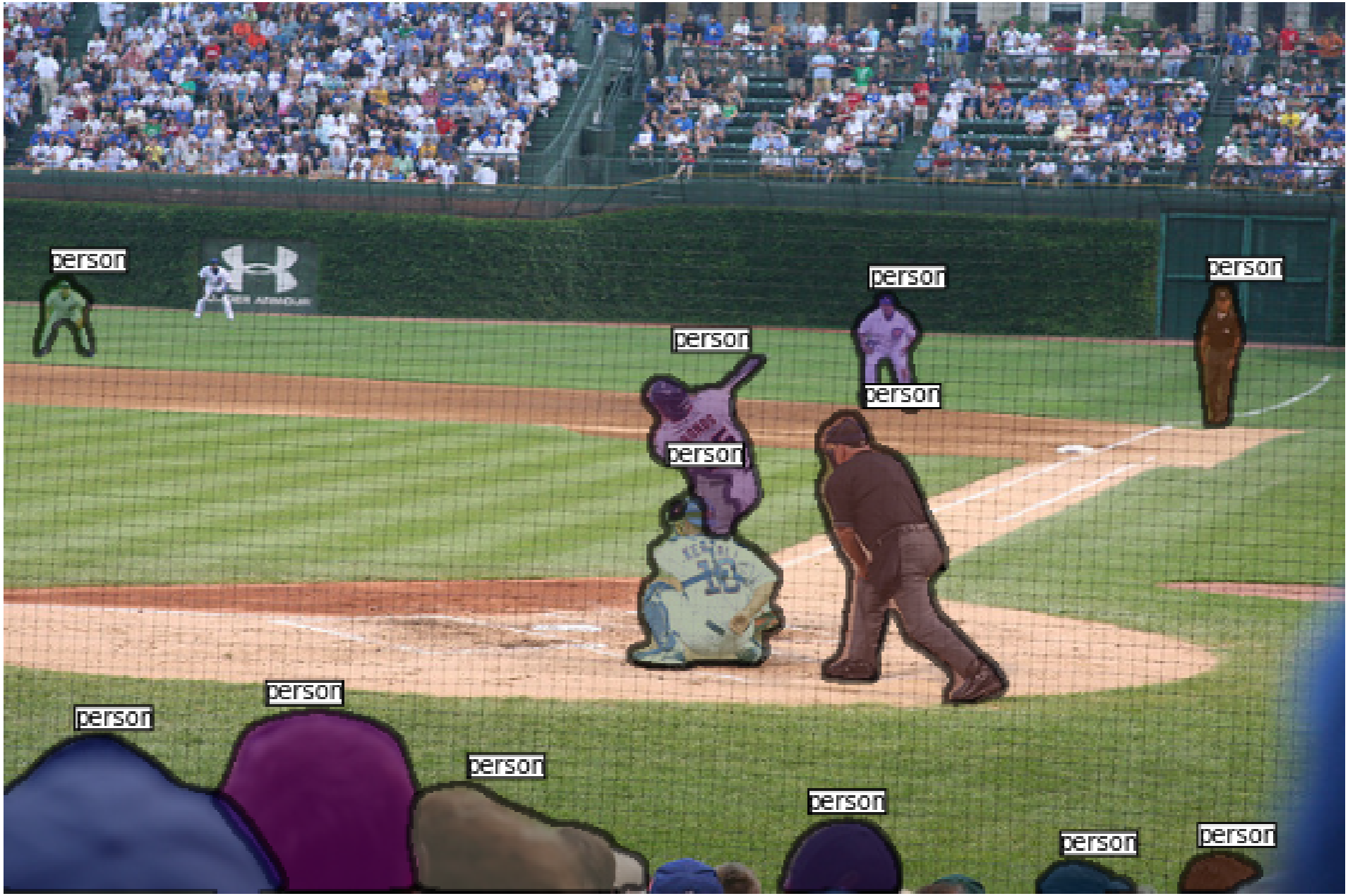
- 2.5 days on 8x4 Kepler GPUs with Elastic Avergaing Stochastic Gradient Descent (EASGD [Zhang, Choromanska, LeCun NIPS 2015]
- "Big Sur"

Results



Results

Results


person
person
person
person
person
person
person
person
person
person
person
person

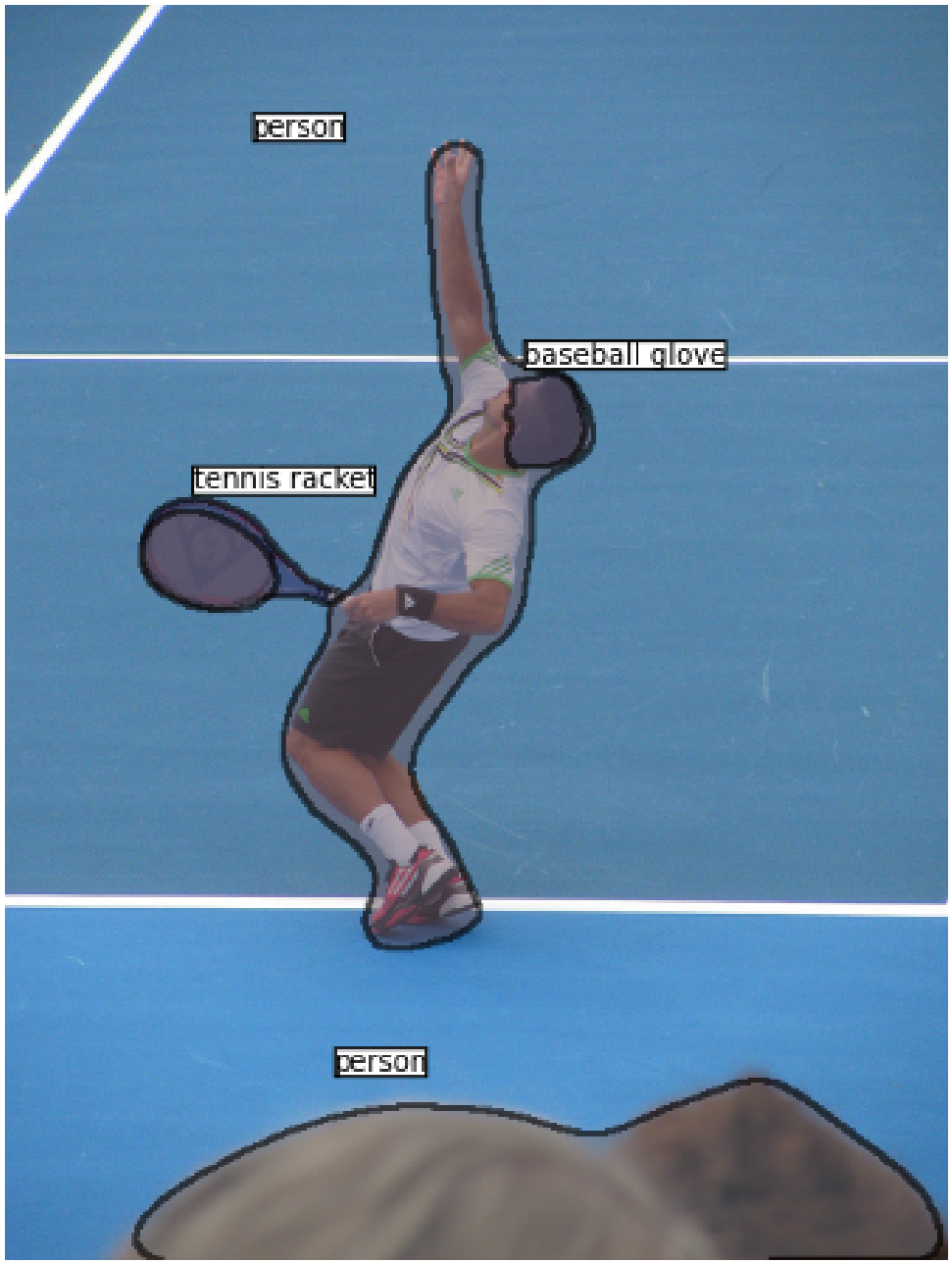
Results

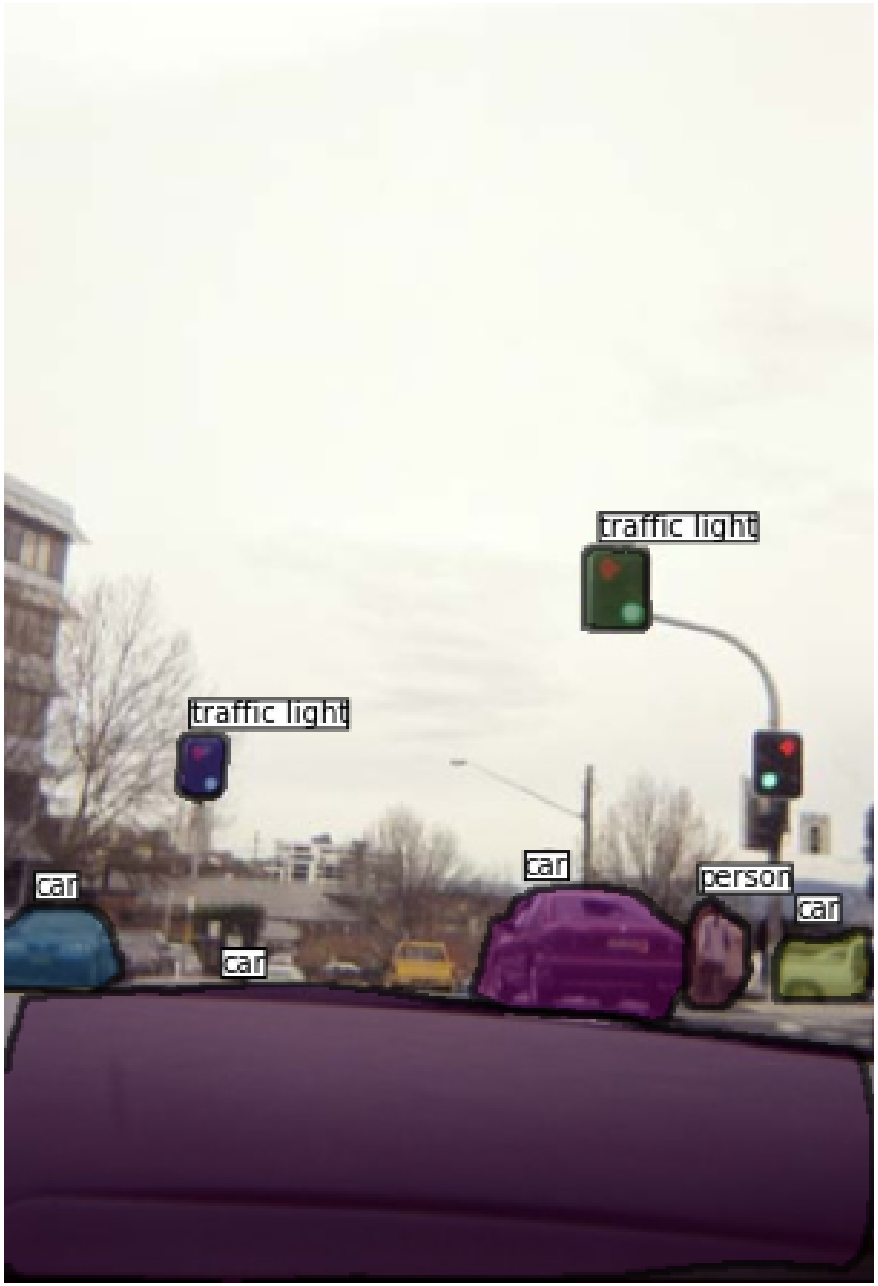
Results

Results


broccoli
broccoli
carrot
carrot
broccoli
broccoli
carrot
broccoli
carrot
broccoli
carrot
broccoli

Mistakes


person
baseball glove
tennis racket
person


traffic light
traffic light
car
car
car
person
car

Results

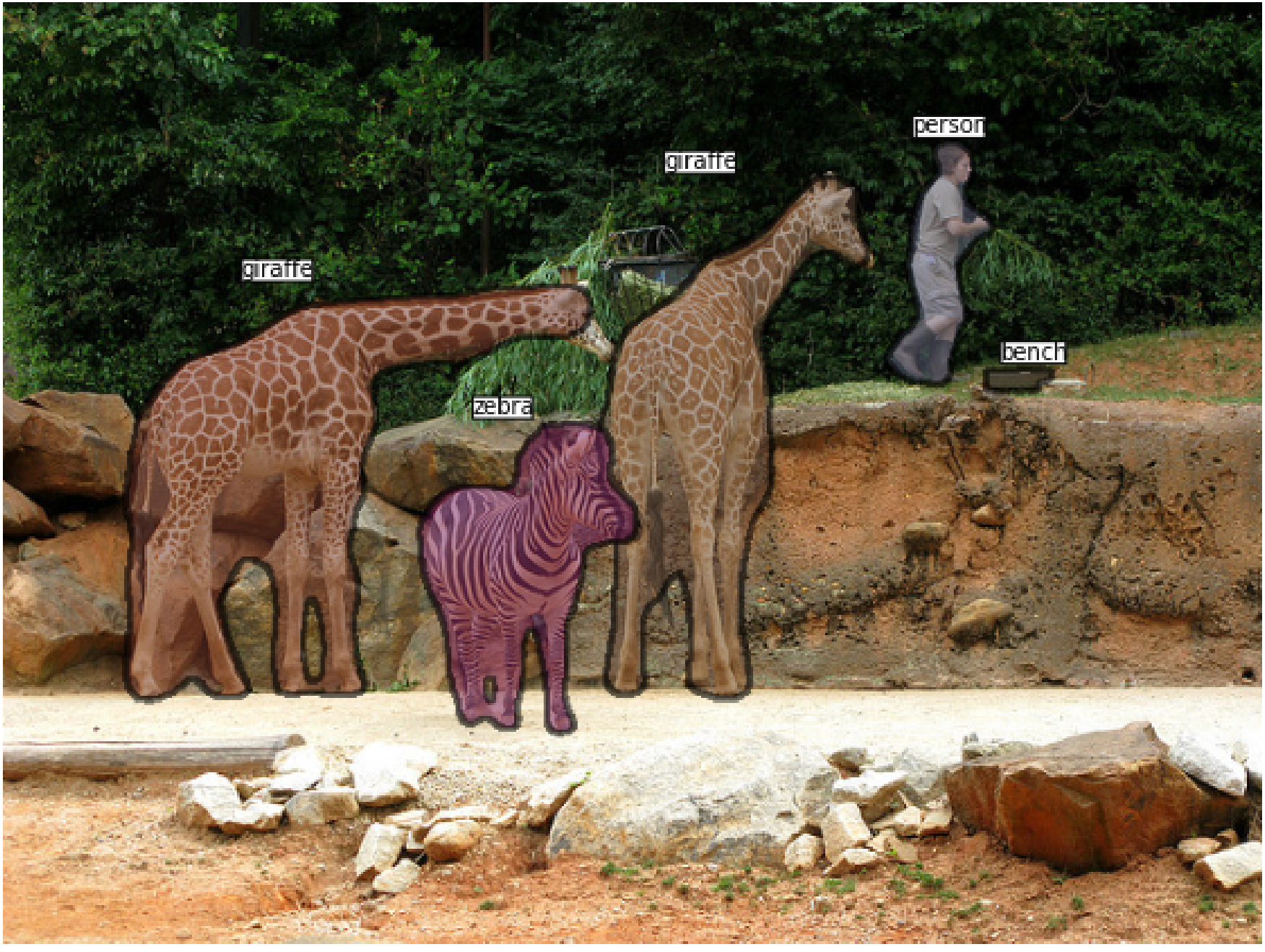
Results


sheep
sheep
sheep
sheep
sheep
sheep
sheep
sheep
sheep
sheep
sheep
sheep
sheep
sheep

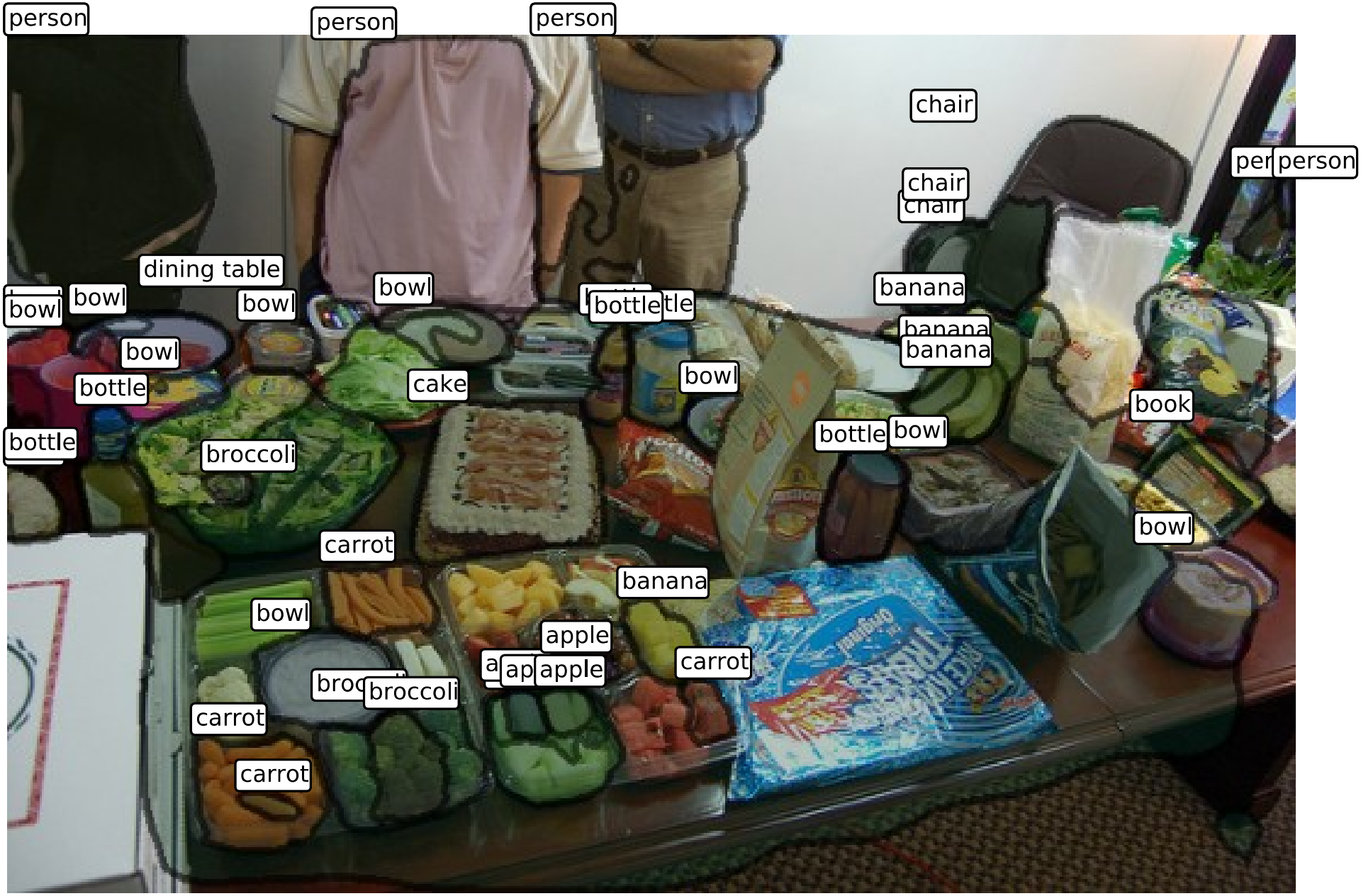
Results


donut
sandwich
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut
donut

Results

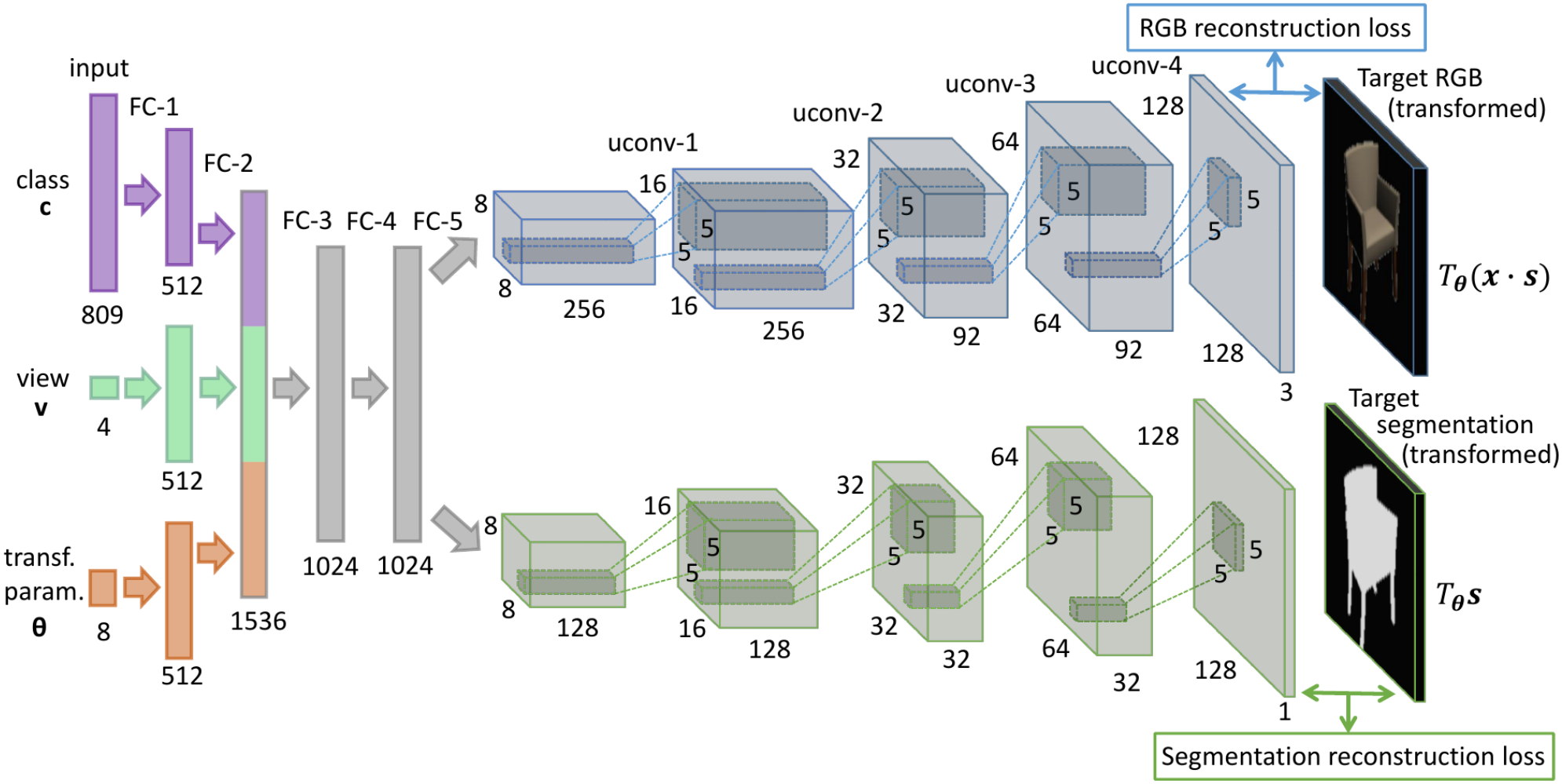


Results


person
person
person
chair
chair
person
dining table
bowl
bowl
bowl
bowl
bowl
bottle
banana
banana
bowl
bottle
cake
bowl
book
bottle
broccoli
bottle
bowl
bowl
carrot
banana
bowl
apple
carrot
broccoli
carrot
carrot

Supervised ConvNets that Draw Pictures

- Using ConvNets to Produce Images
- [Dosovitskyi et al. Arxiv:1411:5928

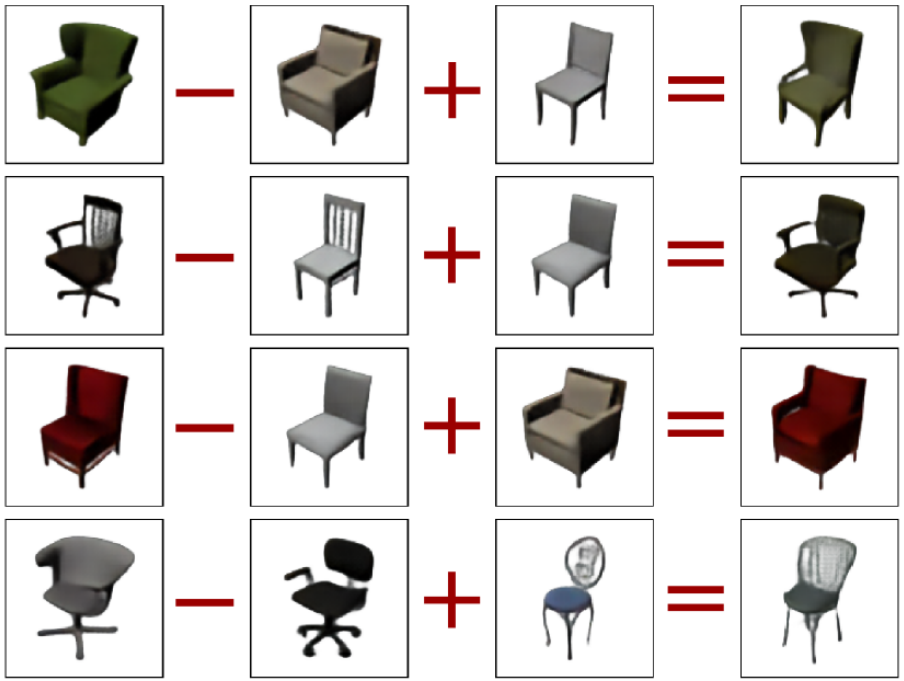


Supervised ConvNets that Draw Pictures

Y LeCun

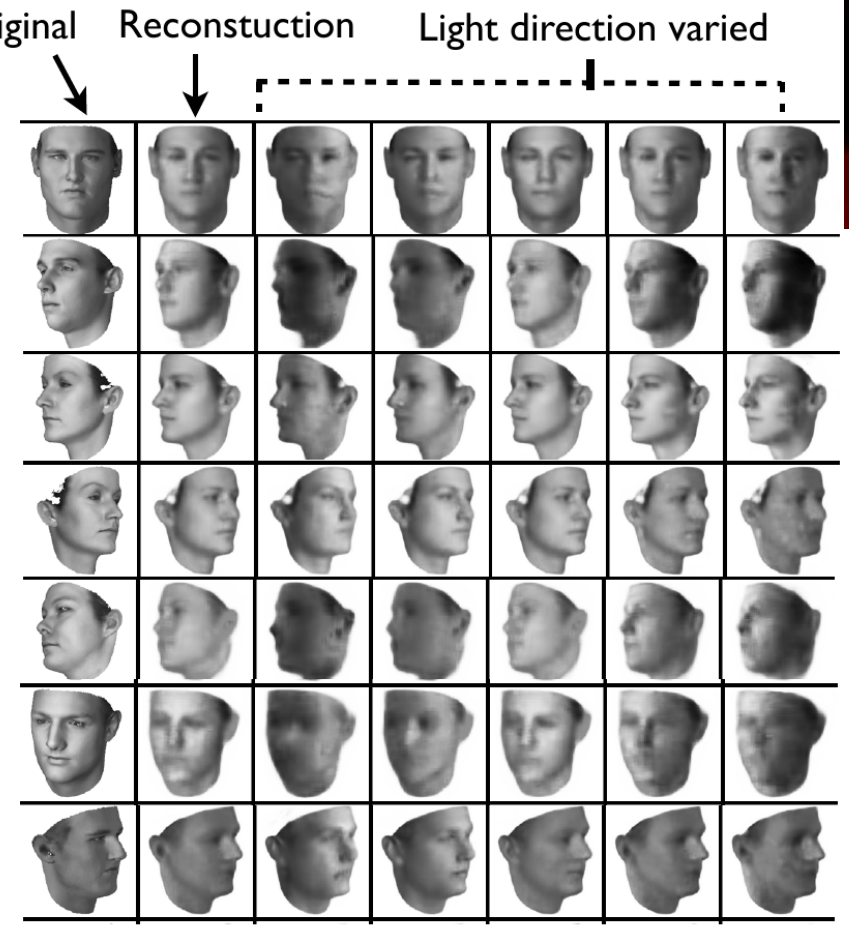
Generating Chairs

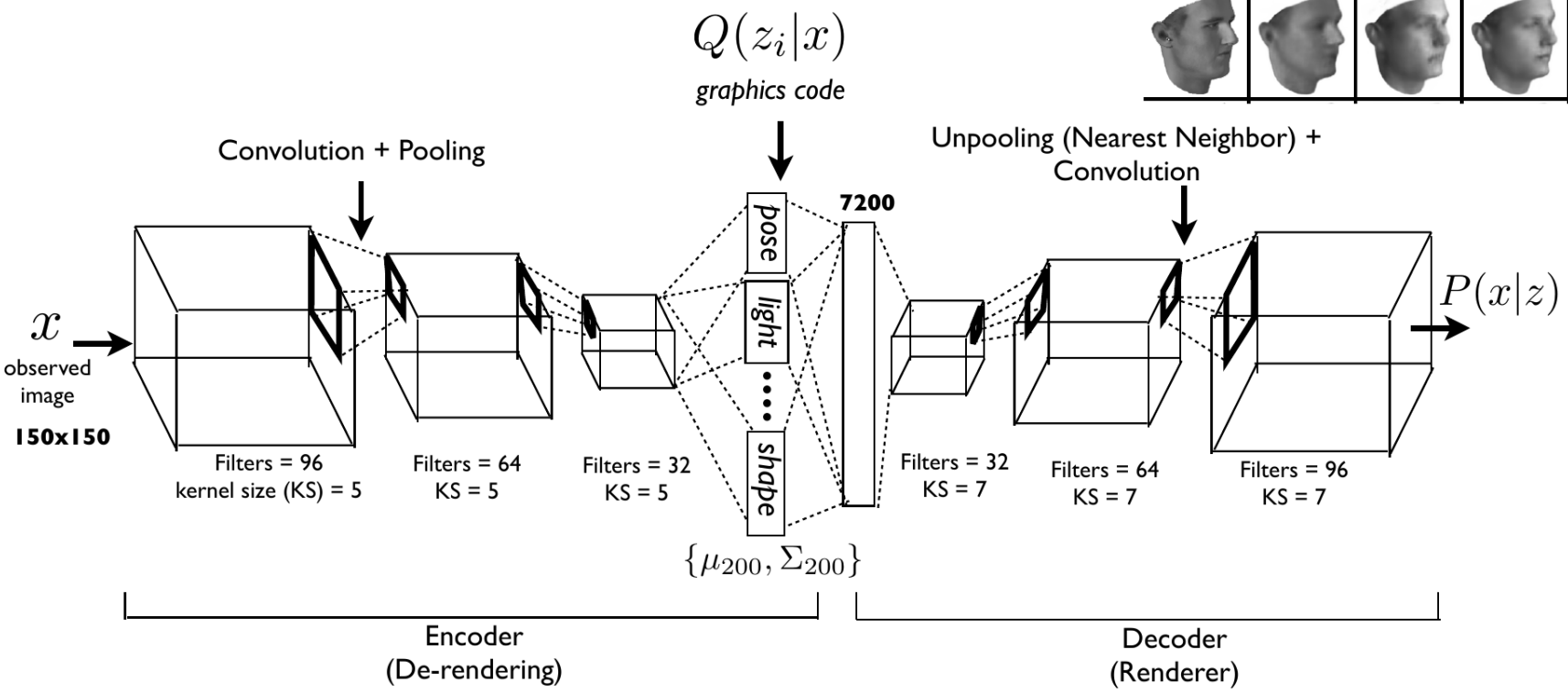
Chair Arithmetic in Feature Space



Convolutional Encoder-Decoder

- Generating Faces
- [Kulkarni et al. Arxiv:1503:03167]

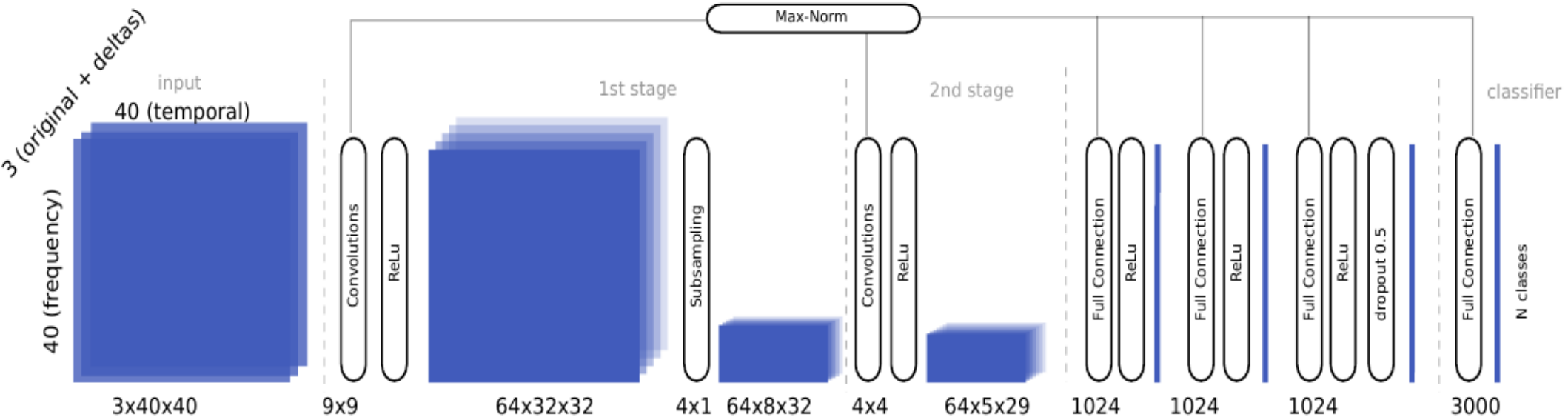





Speech Recognition
With
ConvNets

Speech Recognition with Convolutional Nets (NYU/IBM)

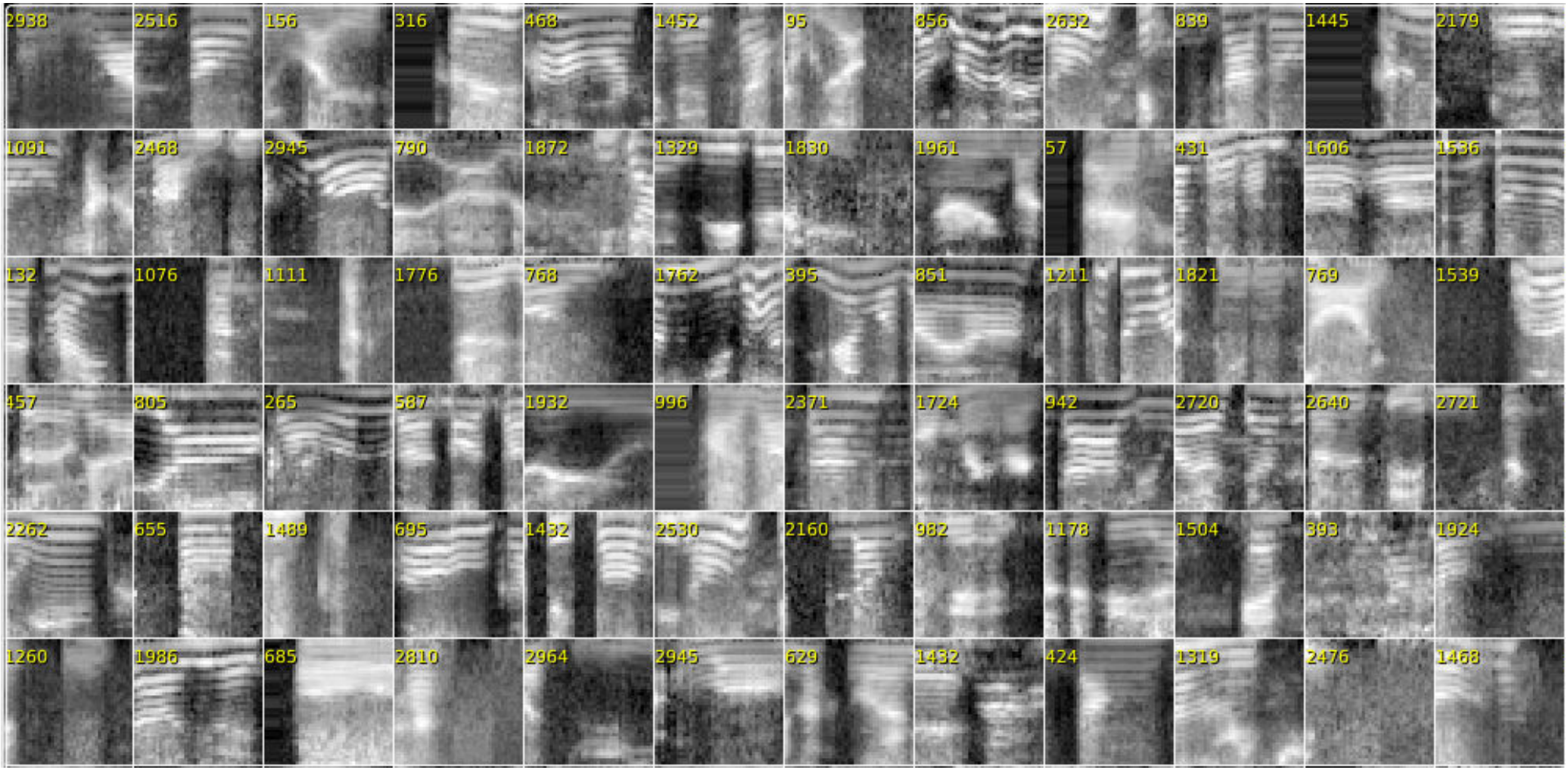
Y LeCun



- Acoustic Model: ConvNet with 7 layers. 54.4 million parameters.
- Classifies acoustic signal into 3000 context-dependent subphones categories
- ReLU units + dropout for last layers
- Trained on GPU. 4 days of training

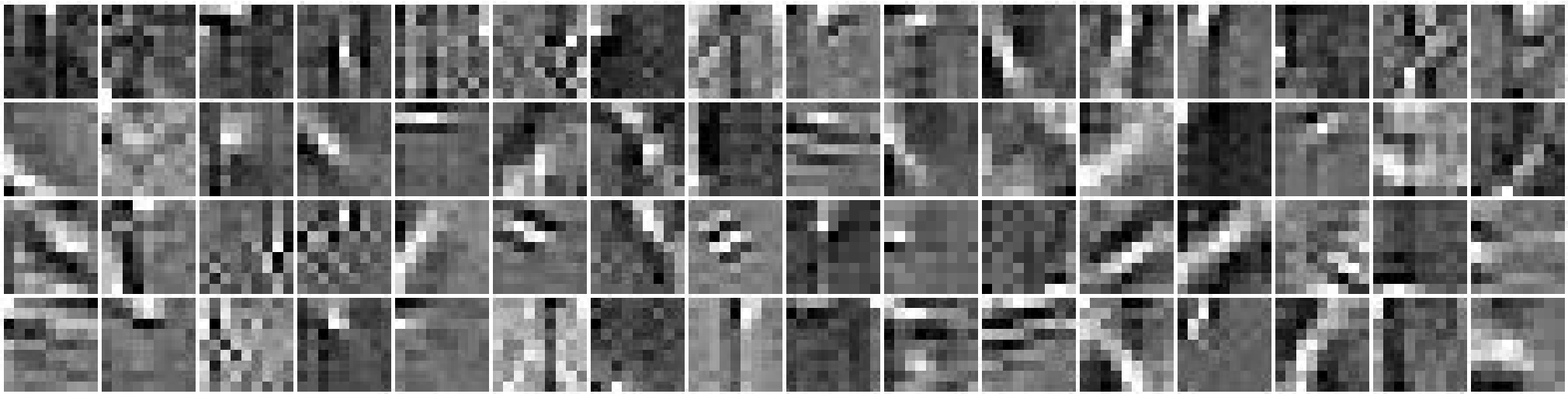
Speech Recognition with Convolutional Nets (NYU/IBM)

- Training samples.
  - 40 MEL-frequency Cepstral Coefficients
  - Window: 40 frames, 10ms each



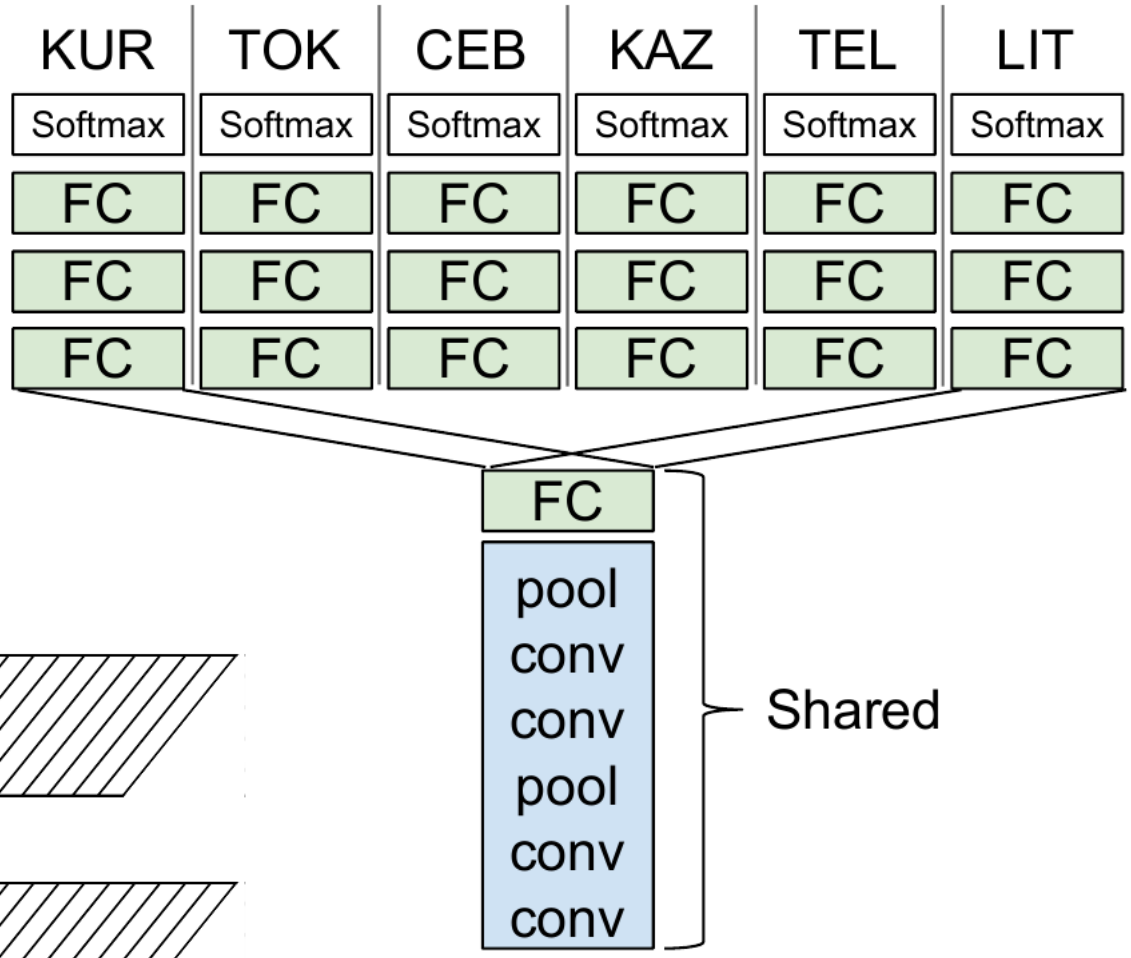
Speech Recognition with Convolutional Nets (NYU/IBM)

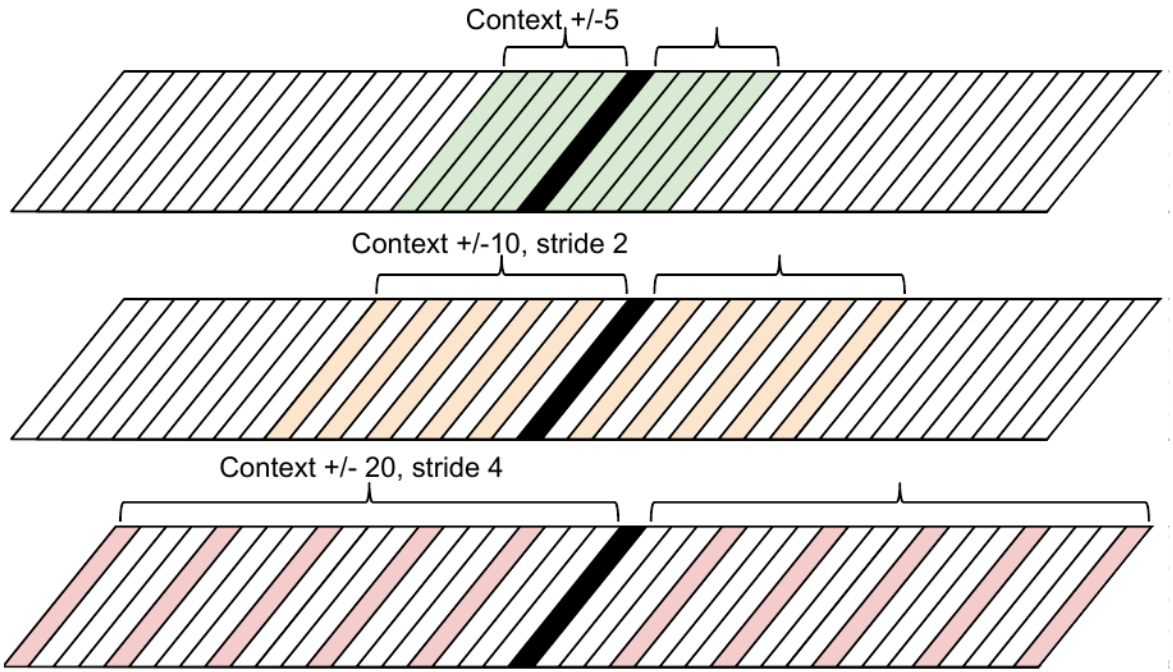
- Convolution Kernels at Layer 1:
  - 64 kernels of size 9x9



Speech Recognition with Convolutional Nets (NYU/IBM)

- Multilingual recognizer
- Multiscale input
  - Large context window






ConvNets are Everywhere
(or soon will be)

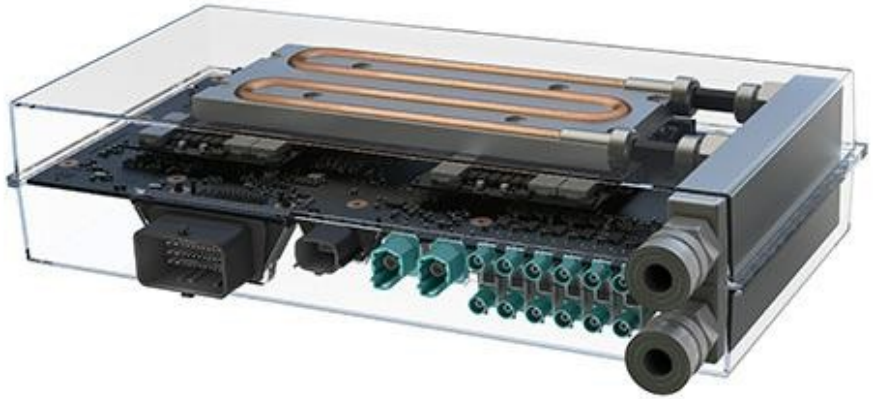
NVIDIA: ConvNet-Based Driver Assistance

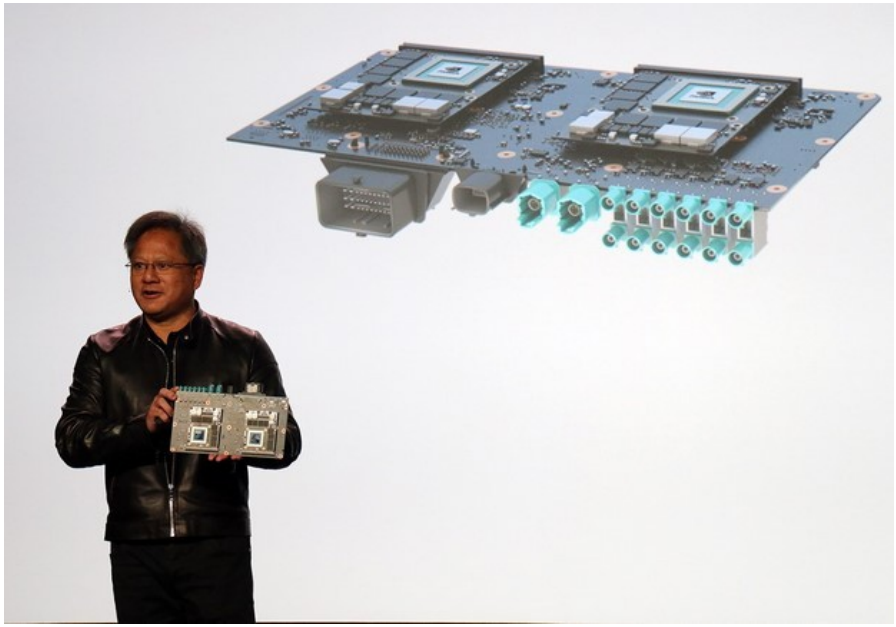
Y LeCun

Drive-PX2: Open Platform for Driver Assistance

Embedded Super-Computer: 42 TOPS

- (=150 Macbook Pros)



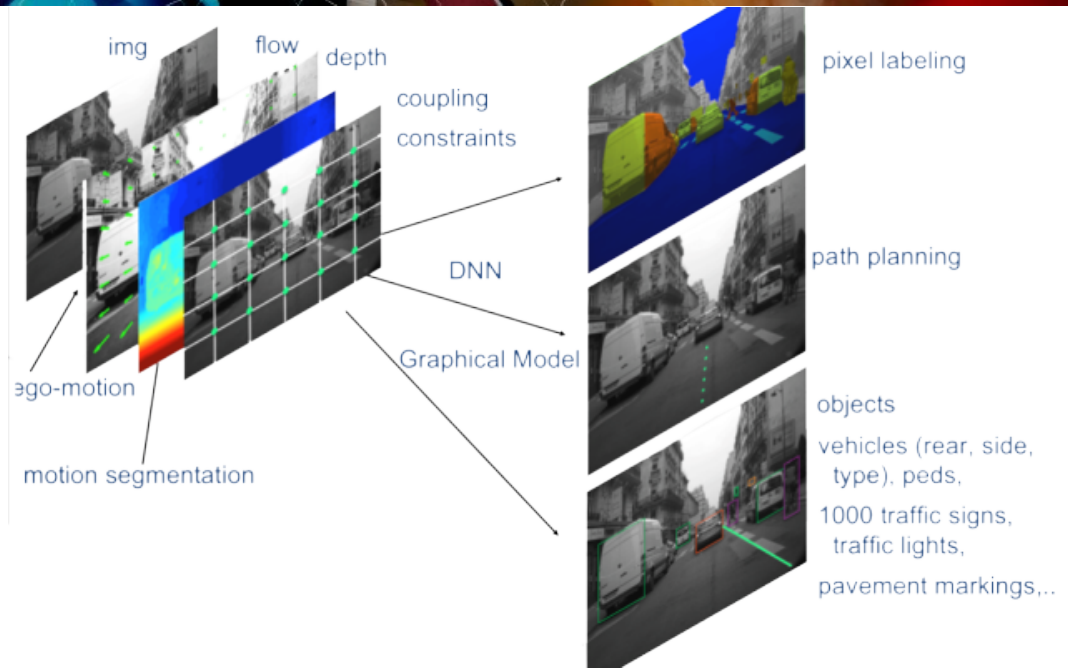


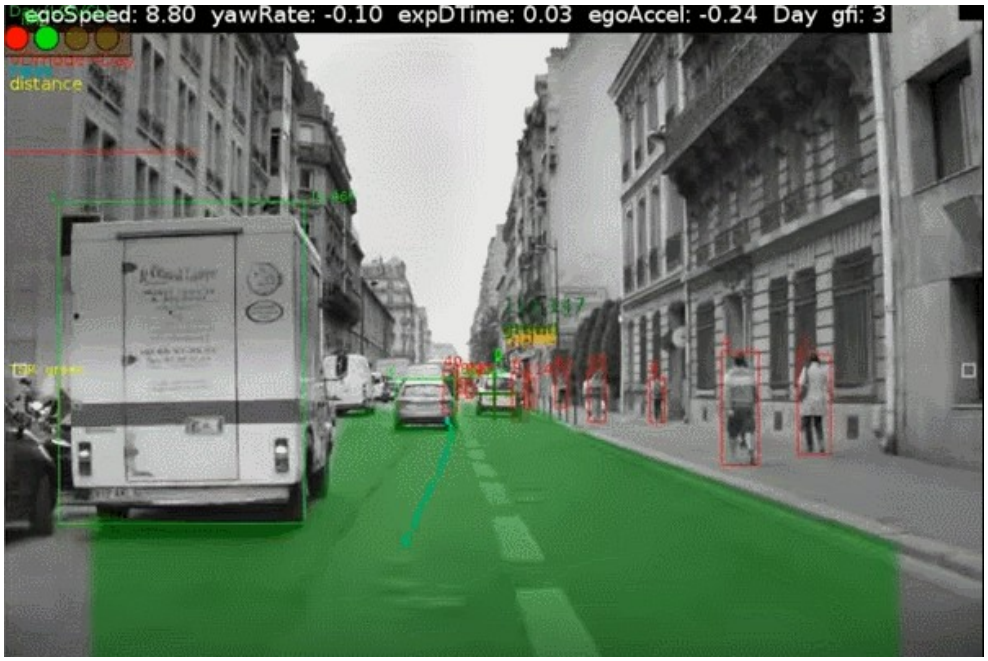
MobilEye: ConvNet-Based Driver Assistance

Y LeCun

Deployed in the latest

Tesla Model S and Model X



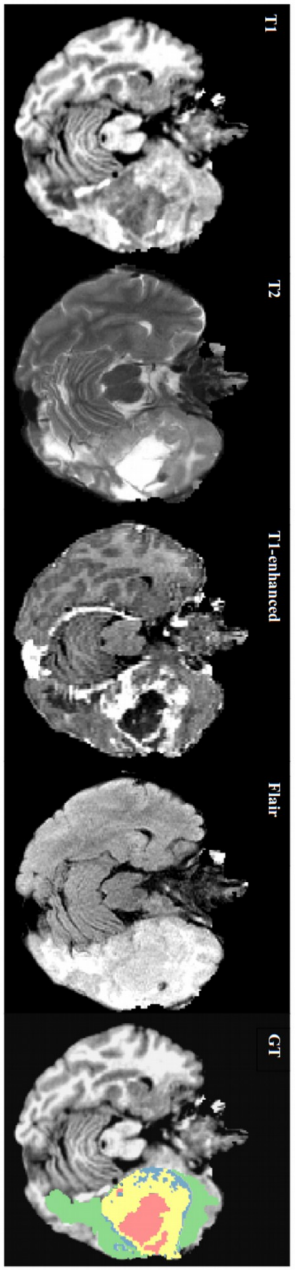


ConvNet in Connectomics [Jain, Turaga, Seung 2007]

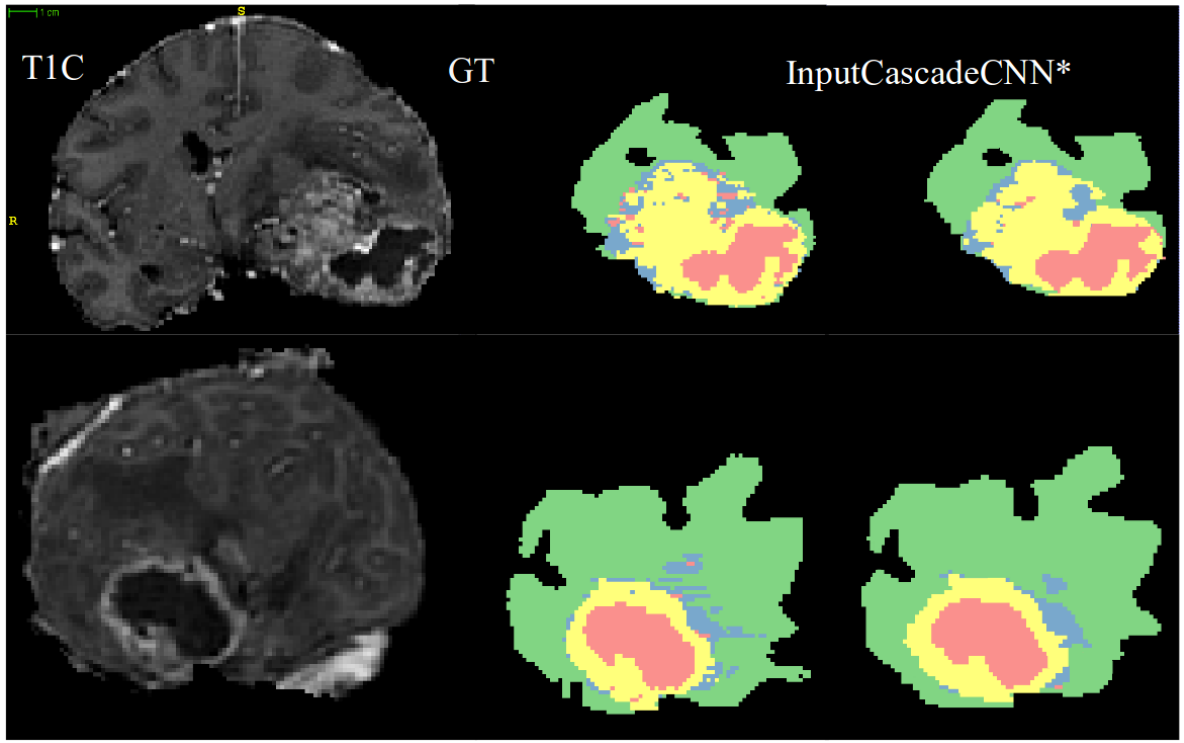
- 3D ConvNet
  - Volumetric
  - Images
- Each voxel labeled as “membrane” or “non-membrane using a 7x7x7 voxel neighborhood
- Has become a standard method in connectomics

VIDEO

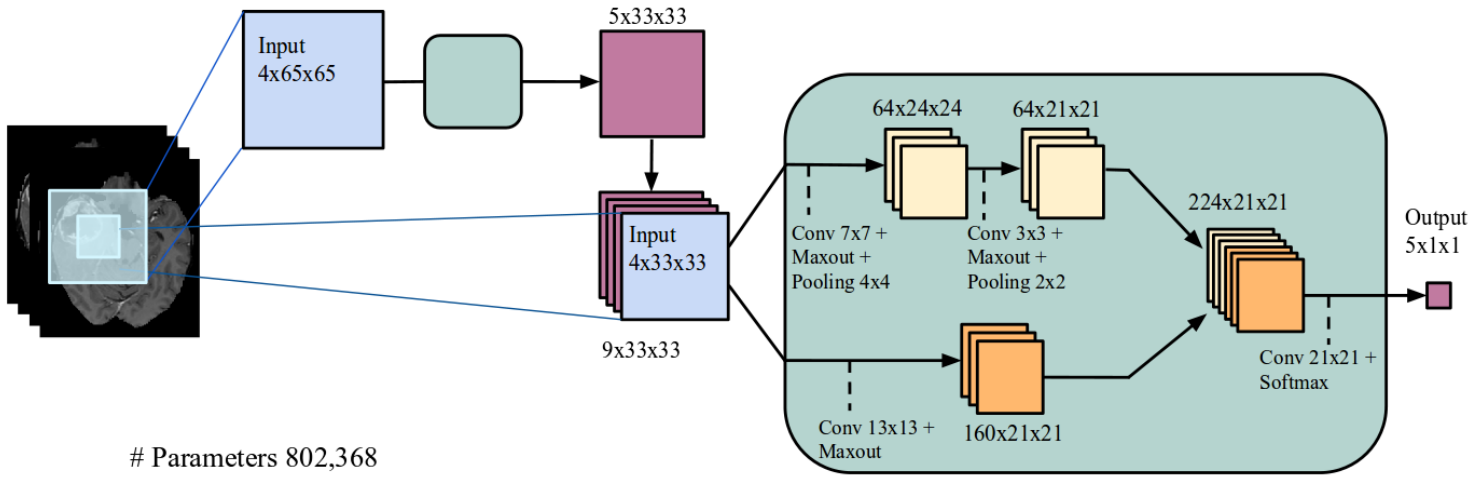
Brain Tumor Detection



- [Havaei et al. 2015]
  - Arxiv:1505.03540
- InputCascadeCNN architecture
  - 802,368 parameters
- Trained on 30 patients.
- State of the art results on BRAT2013



edema, enhanced tumor, necrosis, non-enhanced tumor.



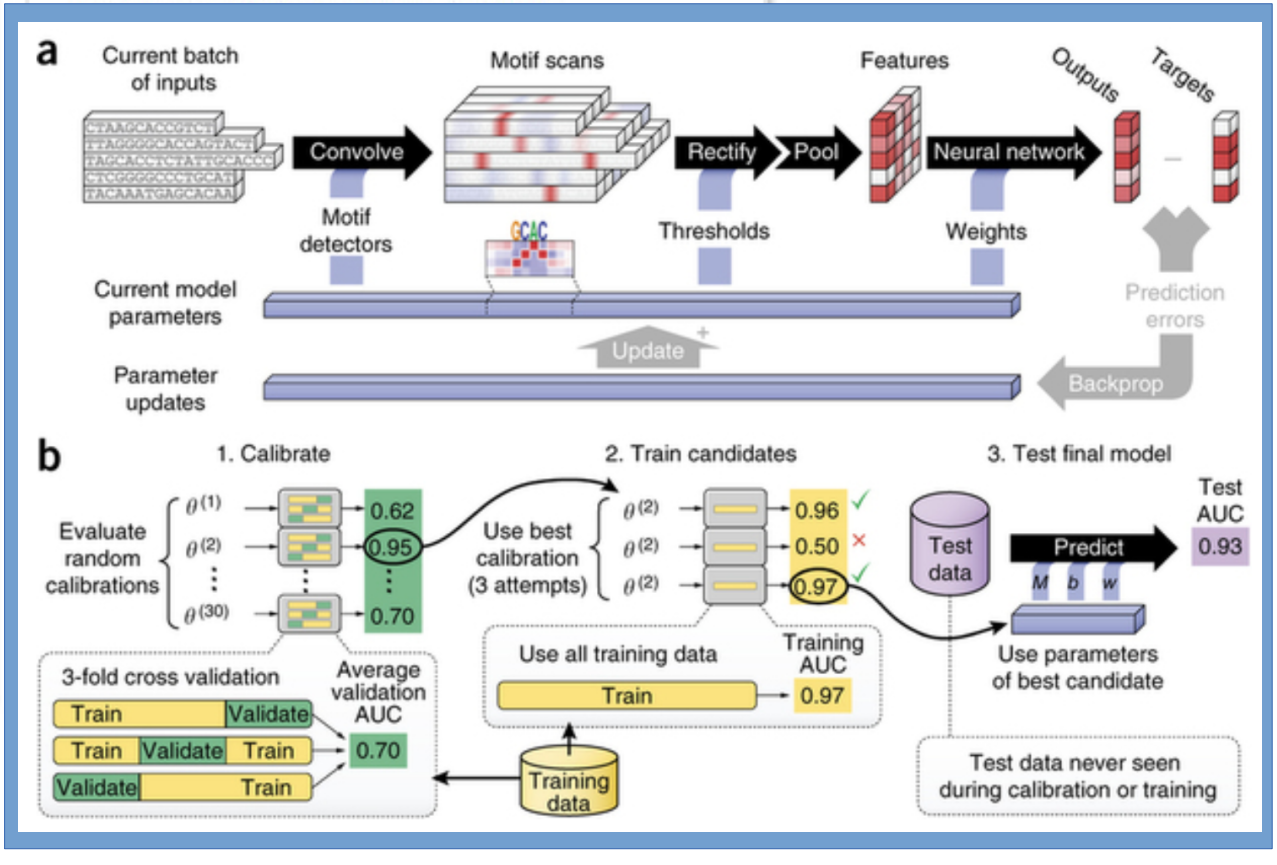
Parameters 802,368

Predicting DNA/RNA – Protein Binding with ConvNets

Y LeCun

"Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning" by B Alipanahi, A Delong, M Weirauch, B Frey,

Nature Biotech, July 2015.



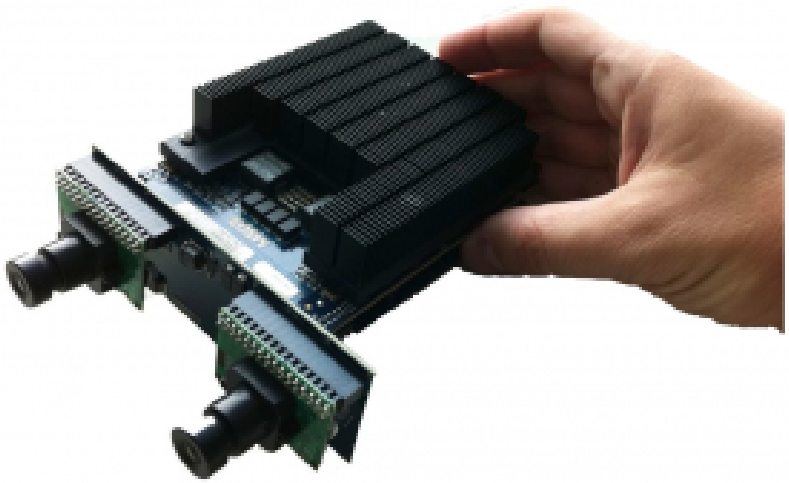
Deep Learning is Everywhere (ConvNets are Everywhere)

- **Lots of applications at Facebook, Google, Microsoft, Baidu, Twitter, IBM…**
  - Image recognition for photo collection search
  - Image/Video Content filtering: spam, nudity, violence.
  - Search, Newsfeed ranking
- **People upload 800 million photos on Facebook every day**
  - (2 billion photos per day if we count Instagram, Messenger and Whatsapp)
- **Each photo on Facebook goes through two ConvNets within 2 seconds**
  - One for image recognition/tagging
  - One for face recognition (not activated in Europe).
- **Soon ConvNets will really be everywhere:**
  - self-driving cars, medical imaging, augemnted reality, mobile devices, smart cameras, robots, toys…..

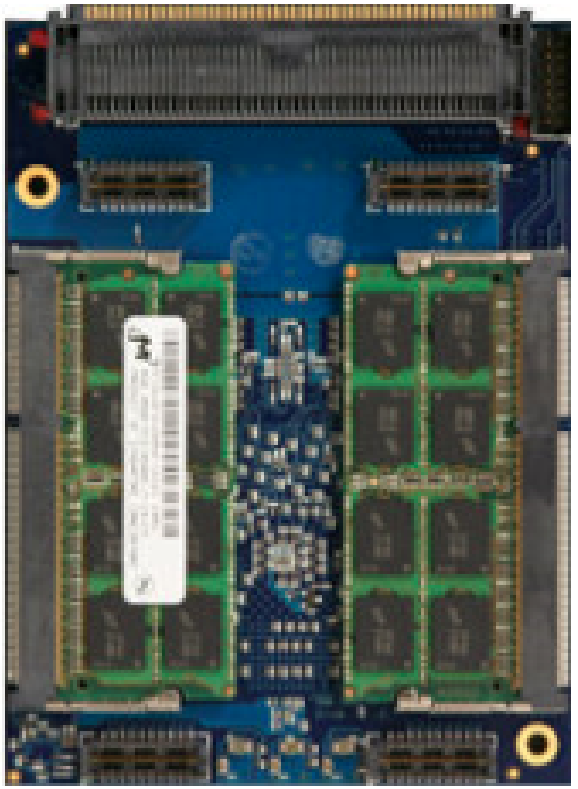

ConvNet
Hardware

NeuFlow architecture (NYU + Purdue)

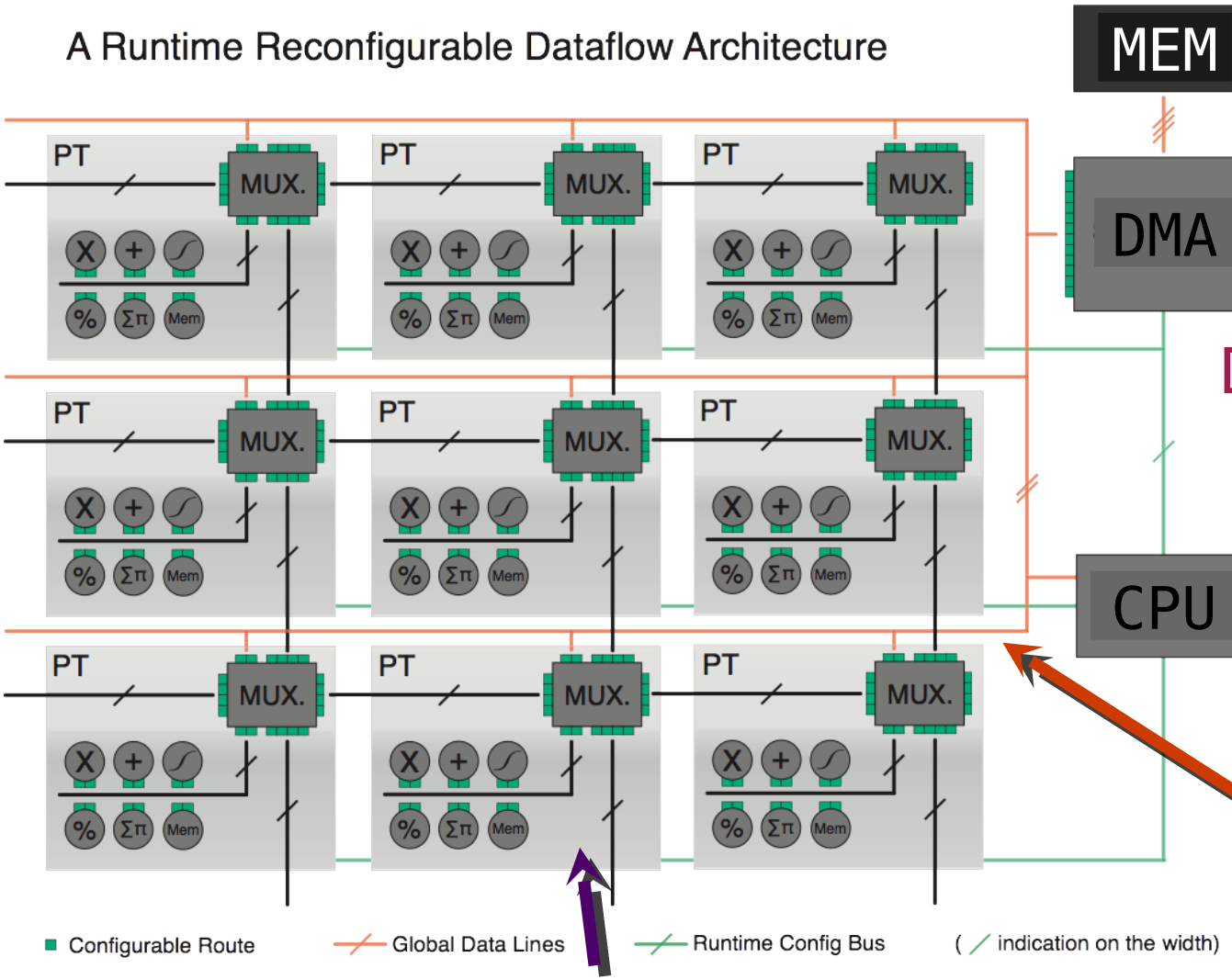
- Collaboration NYU-Purdue: Eugenio Culurciello's e-Lab.
- Running on Picocomputing 8x10cm high-performance FPGA board
  - Virtex 6 LX240T: 680 MAC units, 20 neuflow tiles
- Full scene labeling at 20 frames/sec (50ms/frame) at 320x240



board with Virtex-6



NewFlow: Architecture



MEM

DMA

Multi-port memory controller (DMA)

[x12 on a V6 LX240T]

CPU

RISC CPU, to reconfigure tiles and data paths, at runtime

global network-on-chip to allow fast reconfiguration

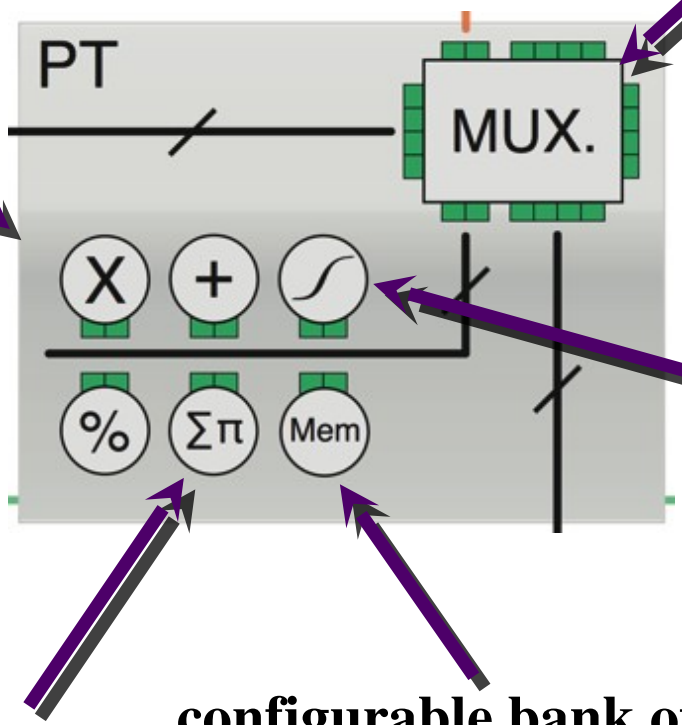
grid of passive processing tiles (PTs)

[x20 on a Virtex6 LX240T]

NewFlow: Processing Tile Architecture

Term-by:-term streaming operators (MUL,DIV,ADD,SUB,MAX)

[x8,2 per tile]



configurable router, to stream data in and out of the tile, to neighbors or DMA ports

[x20]

configurable piece-wise linear or quadratic mapper

[x4]

full 1/2D parallel convolver with 100 MAC units

[x4]

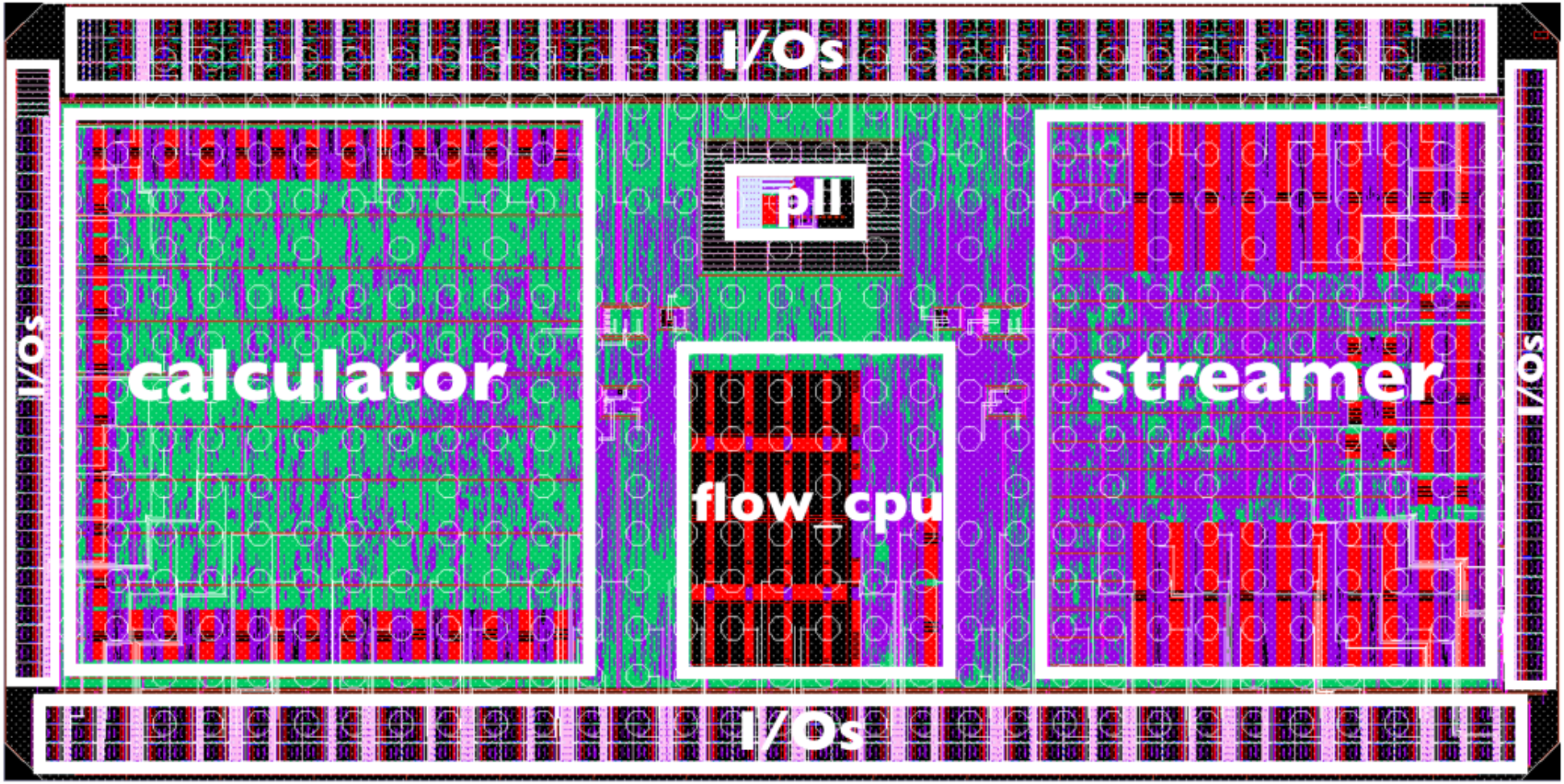
configurable bank of FIFOs , for stream buffering, up to 10kB per PT

[x8]

[Virtex6 LX240T]

NewFlow ASIC: 2.5x5 mm, 45nm, 0.6Watts, >300GOPS

- Collaboration Purdue-NYU: Eugenio Culurciello's e-Lab
- Suitable for vision-enabled embedded and mobile devices
- (but the fabrication was botched...)



[Pham, Jelaca, Farabet, Martini, LeCun, Culurciello 2012]

Deep Learning Applications Today = ConvNets

- **Practical/deployed applications of deep learning:**
  - Almost exclusively ConvNet trained with Backprop
  - A few recurrent nets used experimentally
  - Applications: image recognition, face recognition, video understanding, speech recognition, natural language understanding.
- **Training**
  - Backprop running on racks of GPUs
- **Production**
  - ConvNets on CPUs, with reduced-precision arithmetics
  - Some are moving to GPUs or FPGA.
- **Scale**
  - People upload 500 million images on Facebook every day
  - All of them go through two ConvNets within 2 seconds
    - One for image tagging, one for face recognition.

ConvNet Hardware Today

- **Training is all done on GPUs (on NVIDIA GPUs)**
  - 4 or 8 GPU cards per node. Many nodes per rack. Many racks.
  - 5 Tflops per card (2014)
  - Training needs performance, programmability, flexibility, accuracy.
  - Power consumption and space is not that important.
- **Many large hardware companies are developing ConvNet accelerators**
  - NVIDIA: evolving from GPU. Embedded applications
    - NVIDIA is betting the farm on deep learning.
  - Intel: ConvNet Fixed Function core. 10X over standard core
  - Mobileye: ConvNet chip for automotive
  - Orcam: low-power ConvNet chip for the visually impaired
  - Qualcomm, Samsung: ConvNet IP for mobile devices
- **Lots of startups are getting into the field**
  - Teradeep, derived from NeuFlow
  - Many others...